Commodore INFO

PRIJS ƒ 7.95 / Bfr. 160

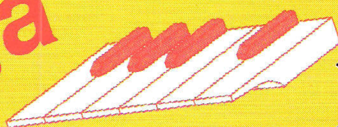


ONAFHANKELIJK BLAD VOOR COMMODORE GEBRUIKERS Jaargang 7, NO.3, ap

LISTINGS:

Engels (64)
Schuifraam (Amiga)

Amiga en MIDI



Extra:

GEOS WORLD

Spellen voor C-64 en Amiga

Amiga op de CeBIT

NIEUW:

PD-hoek voor Amiga

Met Tips en Trucs 64, Oud van Goud, graphics op de 64 en listings voor de 64

COLOFON

Commodore INFO is een uitgave van:
Sala Communications
Postbus 43048
1009 ZA Amsterdam

Uitgever:
V.Sharfman

Redactie:
ir. L. Sala (hoofdredacteur),
drs. J. Boers (eindredacteur),
H. Smeenk, drs. U. Schuurmans,
W.A. Scheer, R.Goudriaan, B. Venema,
P. Boncz, MGCC/Johan&Johan,
Michel de Boer, Hylke Sprangers
telefoon: 020-228871

Produktie:
drs. H. Zoete,
J. Broekhuizen,
M. van Zijl

Advertentie-exploitatie:
ing. V. Sala, D. van Vlijmen
telefoon: 020-273198

Abonnementen en administratie:
Sandra van der Meijden en Marjo Jansen
telefoon: 020-248006

Vragen betreffende abonnementen ontvangen wij bij voorkeur schriftelijk, met meesturen van het omslagetiket.

Vragen over artikelen:
Voor vragen en opmerkingen over artikelen of anderszins graag een briefje aan de redactie (zie boven).

Listingtelefoon:
02155-25162
(ma: 17.00-21.00 u)

Illustraties:
Ben van Mierlo

Cover:
Glass (PD)

Imagesetting IFF:
3Gitaal, Amsterdam

Zetwerk & druk:
NDB, Zoeterwoude

Distributie:
In Nederland: Betapress, Gilze
In België: AMP, Brussel

ISSN: 0169-3085



Inhoud

Redactioneel

Het eerste nummer van Commodore Info dateert van 1983 en we zijn ondertussen zo'n 7 jaargangen verder. Toch ziet het er naar uit, dat we binnenkort naar een andere vorm overgaan. Ondanks de steun van zeer vele trouwe abonnees en wat bedrijven, die de Commodore-markt trouw zijn gebleven, is het draagvlak voor het blad de afgelopen jaren een stuk smaller geworden. Momenteel overwegen we een andere goedkopere produktiewijze om het voortbestaan van het blad nog enige tijd te kunnen garanderen. Ons uitgangspunt hierbij is dat u als lezer niets te kort zal komen.

Zoals we in het vorige nummer al wisten te melden, is het blad Amiga Wereld voortijdig ter ziele gegaan. Het is dan ook niet vreemd dat wij een stapje terug doen.

Commodore Info was het eerste blad, waarmee ik als uitgever de markt op ging en is ook de basis geweest van de verdere bladenreeks, zoals Computer Info, PC Business Info, maar ook de al vergeten Atari Info en Schneider Info. Het doet dus een beetje pijn deze uitgave in een ander jasje te gieten, mijn oudste kind is echter toe aan een transformatie. Dus met wat weemoed groet ik u,

Luc Sala

Abonnement:

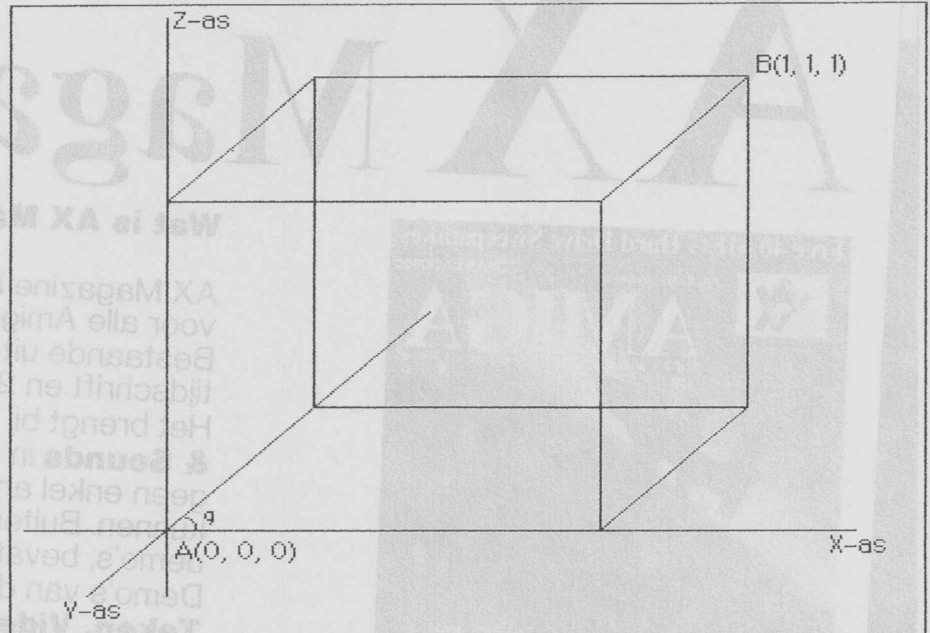
Voor 5 nummers *f* 35,- of BFR. 610 tot het einde van het jaar.
Betaling op giro op giro 4985259 (België: BBL nr. 310050602562) t.n.v. SAC/Commodore-Info.
Oude nummers kunt u alleen krijgen bij vooruitbetaling van *f* 6,75 op de bovenstaande rekening. Zie hiervoor de bon op pagina 24.
Ook telefonisch bestellen is mogelijk. Bel GRATIS 06-02242222 (teleservice) elke dag van 20.20 uur (dus ook in het weekend). België: 115555, dagelijks tot 22.00 uur. Deze nummers zijn allen bedoeld voor de opgave van nieuwe abonnementen.
Opzegging dient schriftelijk te geschieden uiterlijk twee maanden voor de aanvang van een nieuwe abonnementsperiode van een jaar.

Graphics op de 64

Deel 9: ComputerKunst

In deel 6 van deze cursus hebben Hylke Sprangers en Michel de Boer een begin gemaakt met het onderwerp ComputerKunst. In die aflevering is aandacht besteed aan het creeëren van leuke twee-dimensionale high resolution figuren en aan bewegende sprite-kunst. In deze aflevering wordt er een extra dimensie aan die ComputerKunst toegevoegd. Deze aflevering is geheel gewijd aan het construeren van drie-dimensionale objecten. De behandelde onderwerpen zullen uitnodigen tot menig uurtje creatief computer-vermaak.

In deel 6 van de cursus 'Graphics op de 64' hebben we ons voornamelijk bezig gehouden met het maken van mooie figuren in het twee-dimensionale vlak. Deze figuren werden allemaal gemaakt in de high resolution-mode. We hebben toen bijvoorbeeld gezien hoe lissajous figuren en sinuskrommen eruitzagen.
In deze aflevering gaan we verder op het gebied van ComputerKunst. We gaan deze keer een dimensie dieper, en experimenteren in de drie-dimensionale ruimte. Zoals u waarschijnlijk al heeft gezien aan de illustraties bij dit artikel, kan dit leiden tot verrassende resultaten. Voorwaarde hiervoor is echter, dat u wel wat geduld heeft. Drie-dimensionale graphics vergen immers een stuk meer rekenwerk dan twee-dimensionale figuren. Al met al zult u echter nooit langer dan een kwartier op het resultaat hoeven te wachten, en in de meeste gevallen heeft u al na 10 minuten resultaat op het scherm.
Het begin van deze aflevering zal wat theoretisch van aard zijn, maar daarna zullen we een aantal leuke programma's geven waarmee u naar hartelust kunt experimenteren.
Over het al dan niet kunst zijn van computer graphics, hebben we in deel 6 van deze cursus al een verhaal opgehangen. Ook deze keer hebben we de aflevering als ondertitel ComputerKunst meegegeven. Feit blijft wel, dat al de figuren die we deze aflevering bespreken, gebouwd zijn op een wiskundige ondergrond. Aan deze discussie zullen we hier verder geen woorden meer vuil maken. Ieder voor zich kan haar of zijn eigen interpretatie geven aan het woord ComputerKunst, en aan dit woord in relatie met het gebodene.



Figuur 1

Simon's Basic

Alle programma's in deze aflevering zijn geschreven in Simon's Basic. De enige instructies die gebruikt worden uit deze uitbreiding zijn HIRES om het hires scherm aan te zetten, en LINE om een lijn te trekken tussen twee punten. U kunt natuurlijk ook een andere uitbreiding nemen, die twee soortgelijke instructies bevat. U moet er dan wel om denken dat de coördinaten (0, 0) zich bij Simon's Basic linksboven bevinden. Bij sommige uitbreidingen zullen de coördinaten (0, 0) zich linksonder op het scherm bevinden. Voor deze uitbreidingen zult u de programma's een klein beetje moeten aanpassen. Verder moet u overal waar {pi} in de listing staat, gewoon de Griekse letter (naast de restore-toets) intikken.

Het 3D-vlak

Zoals gezegd, gaan we ons in dit artikel bezig houden met drie-dimensionale (3D) figuren. Hiervoor gaan we eerst nog even terug naar de twee-dimensionale (2D) figuren. Als er in een 2D-ruimte wordt gewerkt, betekent dit, dat de plaats van elk punt in de ruimte wordt beschreven door twee coördinaten: een x- en een y-coördinaat. In deze ruimte werken we dan ook met twee assen, een x- en een y-as, die loodrecht op elkaar staan. Een functie in deze ruimte is niets anders dan een (oneindige) verzameling punten. We bekijken hiertoe nog eens een lissajous figuur:

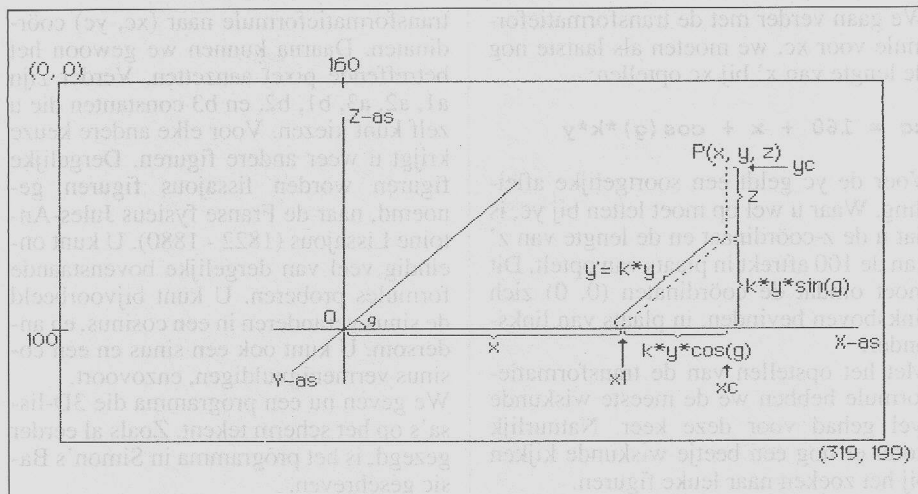
```
x = 40 * sin(2*t)
y = 30 * cos(3*t)   met 0 <=
     t <= 360
```

De x- en y-coördinaat worden allebei beschreven als functie van t. Als we de t laten lopen van 0 to 360 graden, dan zullen we steeds verschillende (x, y) coördinaten

krijgen. Bij het high resolution scherm heeft elk punt (pixel) ook een x- en een y-coördinaat. Oftewel, het hires-scherm is een voorbeeld van een 2D-vlak. De coördinaten (0, 0) bevinden zich linksboven, en de coördinaten (319, 199) rechtsonder. Het is dan ook niet moeilijk om het lissajous figuur op het scherm te zetten. Elke coördinaat van de lissajous kan immers opgevat worden als een beeldscherm-coördinaat. Er bestaat een 1-1 relatie tussen de punten van de lissajous en het hires-scherm.
Voor een 3D-figuur plakken we er nog een dimensie aan. De plaats van elk punt in de drie-dimensionale ruimte wordt beschreven door drie coördinaten, te weten een x-, een y- en een z-coördinaat: (x, y, z). In deze ruimte hebben we dus ook nog een z-as, naast de x- en de y-as. In figuur 1 kunnen we een voorbeeld zien van een figuur in een 3D-ruimte. De x-as loopt horizontaal, de z-as loopt verticaal (naar boven), en de y-as loopt schuin naar achter. In deze ruimte is als voorbeeld een kubus getekend, met ribbe 1. Zo heeft punt A de coördinaten (0, 0, 0) en punt B de coördinaten (1, 1, 1). Merk op dat de y-as onder een hoek van 45 graden met de x-as is getekend. Deze hoek zullen we in het vervolg hoek g noemen. De positieve y-as is in dit figuur naar achter getekend.

Van 3D naar 2D

We hebben nu gezien dat elk punt in de 3D-ruimte 3 coördinaten heeft: (x, y, z). Nu willen we een 3D-figuur op het scherm laten zien. Het probleem dat zich nu opwerpt, is dat elke pixel op het scherm slechts twee coördinaten heeft. Dit betekent, dat we niet zoals bij 2D-figuren het pixel met de bijbehorende coördinaten aan kunnen zetten. We moeten op de een of andere manier een projectie zien te vinden van een 3D-ruimte op een 2D-vlak. Oftewel, we moeten een transformatie-formule vinden, die een punt met drie coördinaten (x, y, z), afbeeldt op een punt met twee coördinaten (xc, yc). Want dan kunnen we gewoon elk punt in het 3D vlak omrekenen naar een punt in het 2D vlak. Vervolgens kunnen we dit punt (xc, yc) gewoon beschouwen als een pixel op het beeldscherm. Dit pixel kunnen we dan gewoon aanzetten.
Het vinden van zo'n transformatieformule lijkt op het eerste gezicht moeilijker dan het in werkelijkheid is. Voor het opstellen van zo'n formule, moeten we echter eerst een goede projectie vinden van de 3D-ruimte op het 2D-vlak. Kijkt u nog eens naar figuur 1. Zoals al eerder opgemerkt, is in dit figuur een 3D-vlak te zien. In dit vlak is een kubus getekend. Nu moet u bedenken dat het plaatje zelf tweedimensionaal is! U kunt, met andere woorden, elk punt in het plaatje aangeven



Figuur 2

met twee coördinaten (xc, yc). Het feit dat het plaatje toch een 3D-figuur afbeeldt, betekent dat we al een geschikte projectie van een 3D-ruimte op een 2D-vlak hebben gevonden. Door de y-as schuin naar achter te tekenen, kijken we een beetje van rechtsboven op het ruimtelijke vlak neer. Het enige wat we nog moeten doen, is de geschikte transformatieformule vinden bij deze projectie.

Transformatieformule

Nog even resumeren. We hebben de volgende projectie van de 3D-ruimte op het 2D-vlak gevonden. We tekenen de x-as horizontaal en de z-as verticaal. De y-as tekenen we schuin naar achter, onder een hoek g (voor g kiezen we straks 45 graden) met de positieve x-as. Als u figuur 1 nog een keer bekijkt, kunt u zien dat de ribben van de kubus die parallel met de y-as lopen, kleiner zijn dan de ribben langs de x- en z-as. De ribben langs de y-as zijn ongeveer twee keer zo klein gemaakt. Voor onze projectie moeten we de y-coördinaten dus steeds verkleinen, en wel met een factor 2.
Bij deze projectie hoort de volgende transformatieformule van (x, y, z) naar (xc, yc):

```
xc = 160 + x + cos(g)*k*y
yc = 100 - z - sin(g)*k*y
```

We zullen hieronder uitleggen hoe we aan deze formule zijn gekomen. De lezer die deze wiskunde liever laat voor wat ze is, kan het volgende stuk, tot het eind van de paragraaf, overslaan.
We gaan de transformatieformule uitleggen aan de hand van figuur 2. Ten eerste zullen we ons richten op de formule voor xc. De oorsprong van de 3D-ruimte (0, 0, 0) willen we graag in het midden van het beeld hebben. Het midden van het beeld heeft de coördinaten (160, 100). De formule voor xc is dan:

```
xc = 160
```

In figuur 2 is het 3D-assenstelsel getekend, als projectie op het 2D-vlak. Het punt (0, 0) bevindt zich hier dan ook linksboven. Het punt (0, 0, 0) is aangegeven met O, en is dus, zoals gezegd, equivalent met punt (160, 100) in het midden van het scherm. In figuur 2 is een punt P getekend met coördinaten (x, y, z). We zullen voor dit punt de coördinaten (xc, yc) berekenen, en aan de hand daarvan de transformatieformule opstellen. De positieve x-as is naar rechts getekend, dus we moeten bij xc nog de x-coördinaat van het punt P optellen:

```
xc = 160 + x
```

We zijn nu aangeland bij het punt dat in de figuur is aangegeven met x1. Dit is echter nog niet de juiste xc-coördinaat. U kunt dit in de figuur ook zien, want de juiste xc staat ook aangegeven, en bevindt zich aan de rechterkant van x1.
We vatten de y-coördinaat van punt P nu even op als een vector. Deze vector noemen we y' (zie figuur). Bedenk nu dat de y-coördinaat van elk punt verkleind moest worden voor de projectie. Als verkleiningsfactor hebben we het getal 2 genoemd. De reciproke verkleiningsfactor noemen we k. (De reciproke verkleiningsfactor is 1/verkleiningsfactor. Dus k=1/2.) De lengte van de vector y' is dus k*y. De richting van de vector is g graden ten opzichte van de positieve x-as. Deze vector y' gaan we nu ontbinden in een x'-en een z'-vector, die respectievelijk parallel met de x- en z-as lopen. De lengte van x' is cos(g)*y' = cos(g)*k*y. De lengte van z' is sin(g)*y' = sin(g)*k*y. Wat we nu dus gedaan hebben, is het opsplitsen van de y-coördinaat in een x- en een z-richting. We hebben dus nu nog 2 coördinaten over, de x- en de z-coördinaat. En die kunnen we makkelijk omzetten naar een xc-en een yc -coördinaat.

We gaan verder met de transformatieformule voor xc. we moeten als laatste nog de lengte van x' bij xc optellen:

```
xc = 160 + x + cos(g)*k*y
```

Voor de yc geldt een soortgelijke afleiding. Waar u wel op moet letten bij yc, is dat u de z-coördinaat en de lengte van z' van de 100 aftrekt in plaats van optelt. Dit moet omdat de coördinaten (0, 0) zich linksboven bevinden, in plaats van linksonder.
Met het opstellen van de transformatieformule hebben we de meeste wiskunde wel gehad voor deze keer. Natuurlijk komt er nog een beetje wiskunde kijken bij het zoeken naar leuke figuren.

Leuke 3D-figuren

We hebben nu een projectie-formule. Dat wil zeggen dat we elk punt uit het 3D-vlak om kunnen zetten naar een punt in het 2D-vlak. Nu moeten we nog leuke functies of figuren verzinnen om te maken in het 3D-vlak. We onderscheiden figuren met een 1-dimensionale, 2-dimensionale en een 3-dimensionale vorm (en dit alles in het 3D-vlak). Een figuur met een 1-dimensionale vorm is natuurlijk de lijn. Omdat rechte lijnen over het algemeen niet zulke mooie graphics opleveren, houden we ons in dit artikel bezig met de kromme lijnen in het 3D-vlak. In deel 6 hebben we lissajous figuren in het 2D-vlak gezien. We gaan nu lissajousfiguren in het 3D-vlak maken.
Als we het over 2-dimensionale figuren hebben, dan hebben we het natuurlijk over vlakken. We gaan dan ook gebogen (onder andere sinusoïde) vlakken maken in de ruimte.
Als voorbeelden van drie-dimensionale figuren noemen we de bol, cilinder, kegel etcetera, eigenlijk zijn dit slechts gebogen vlakken, en dus ook 2-dimensionale figuren. We zullen deze toch apart behandelen, omdat het programma om ze te maken, verschillend is met het programma om gebogen vlakken te tekenen.

3D-lissajous figuren

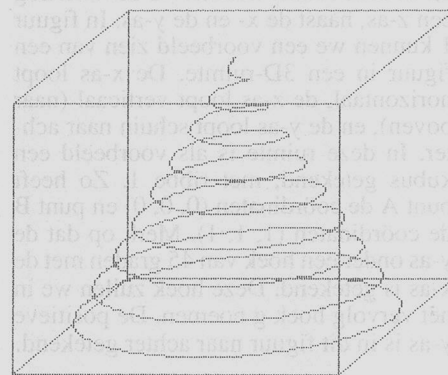
Zo simpel als het is om krommen te maken in het 2D-vlak, zo simpel is het om krommen te maken in het 3D-vlak. We plakken er letterlijk en figuurlijk een dimensie bij. We hebben het dan over krommen met bijvoorbeeld de volgende parametervorm:

```
x = a1 * sin(b1 * t)
y = a2 * sin(b2 * t)
z = a3 * cos(b3 * t)
```

De x-, y- en z-coördinaat zijn alle afhankelijk van een parameter, de t. Als we de t laten lopen van 0 tot 360 graden, worden er steeds andere coördinaten (x, y, z) verkregen. Deze zetten we simpel om met de transformatieformule naar (xc, yc) coördinaten. Daarna kunnen we gewoon het betreffende pixel aanzetten. Verder zijn a1, a2, a3, b1, b2, en b3 constanten die u zelf kunt kiezen. Voor elke andere keuze krijgt u weer andere figuren. Dergelijke figuren worden lissajous figuren genoemd, naar de Franse fysicus Jules-Antoine Lissajous (1822 - 1880). U kunt oneindig veel van dergelijke bovenstaande formules proberen. U kunt bijvoorbeeld de sinus veranderen in een cosinus, en andersom. U kunt ook een sinus en een cosinus vermenigvuldigen, enzovoort.
We geven nu een programma die 3D-lissa's op het scherm tekent. Zoals al eerder gezegd, is het programma in Simon's Basic geschreven.

```
100   rem 3d lissajous
110   c=.5*cos(.25*{pi}):s=.5
      *sin(.25*{pi})
150   r=70:hires 0,1
160   for i=0 to 11:read x,y,z
161   gosub 1000:x1=xc:y1=yc
162   read x,y,z:gosub 1000
163   line x1,y1,xc,yc,1:next
164   data -1,-1,-1,1,-1,-1
165   data -1,-1,-1,-1,1,-1
166   data -1,1,-1,1,1,-1
167   data 1,1,-1,1,-1,-1
168   data -1,-1,-1,-1,-1,1
169   data 1,-1,-1,1,-1,1
170   data 1,1,-1,1,1,1
171   data -1,1,-1,-1,1,1
172   data -1,-1,1,1,-1,1
173   data -1,-1,1,-1,1,1
174   data 1,1,1,1,1,1
175   data 1,1,1,1,-1,1
180   for t1=0 to 360 step 1
185   t=t1*{pi}/180
190   x=sin(4*t)
191   y=cos(4*t)
192   z=cos(5*t)
200   gosub 1000
220   if t1=0 then x1=xc:y1=yc
230   line x1,y1,xc,yc,1
250   x1=xc:y1=yc:next
260   goto 260
1000  xc=int(160+r*x+r*c*y)
1010  yc=int(100-r*z-r*s*y)
1020  return
```

In regel 110 worden er twee waarden uitgerekend voor de variabelen C en S. Als u nog even aan de transformatieformule terugdenkt, dan wordt de y-coördinaat voor xc en yc vermenigvuldigd met respectievelijk k*cos(g) en k*sin(g). Deze twee waarden worden in regel 110 al berekend. Voor k hebben we 0.5 gekozen (de y-coördinaat wordt dan twee keer zo klein) en voor g 45 graden. De computer rekent echter in radialen. Graden kunnen omgerekend worden naar radialen door te vermenigvuldigen met {pi}/180. 45 graden is zo 0.25*{pi} radialen. Om de lissajous in duidelijk perspectief te laten zien, tekenen we eerst een kubus op het beeld. De ribben van deze kubus hebben de lengte 2*r. In regel 150 wordt een waarde aan deze r toegekend. De kubus staat echter niet (zoals in figuur 1) met een hoek in de oorsprong, maar staat om de oorsprong heen, de oorsprong bevindt zich dus midden in de kubus. Het hoekpunt linksonder heeft zo de coördinaten (-r, -r, -r) en rechtsboven (r, r, r). Deze kubus wordt in de regels 160 tot 163 getekend, nadat in regel 150 het hires is aangezet. In de regels 164 tot 175 staat de data die nodig zijn voor het tekenen van de kubus.
In regel 180 wordt de loop van 0 tot 360 geprogrammeerd. Alvorens de formule voor de lissa in te vullen, moeten de graden nog omgerekend worden naar radialen, in regel 185. In de regels 190 tot 192 worden de coördinaten uitgerekend voor een punt van de lissajous. In de subroutine op regel 1000 staat de transformatieformule. Het programma werkt zo, dat het steeds het punt onthoudt dat de vorige keer berekend is. Dit punt wordt bewaard in (x1, y1). Nu wordt er steeds een lijn getrokken tussen twee opeenvolgende punten. Daardoor wordt er een vloeiende lissajous figuur verkregen. Tijdens de eerste iteratie is er echter geen vorig punt berekend (x1, y1 ongedefinieerd). Dit uitzonderingsgeval wordt opgevangen in regel 220. Als vaste vergrotingsfactor wordt er steeds r gekozen. Zo blijft de lissajous steeds binnen de getekende kubus. In **figuur 3** staat een lissajous figuur dat met bovenstaand programma is gemaakt. In **figuur 5** staan nog leuke functies om uit te proberen. Deze functies moet u dan in de regels 190-192 zetten.



Figuur 3

Vlakken in de ruimte

Het maken van gebogen vlakken in de ruimte is haast hetzelfde als het maken van lissajous figuren. Als eerste moet er een vergelijking worden opgesteld van het vlak dat getekend moet worden. Als voorbeeld nemen we de onder de naam 'zadelvlak' bekend staande functie:

```
z = x*x - y*y
```

Voorbeeld 3D lissajous funkties		
x	y	z
x = sin(t)	y = sin(2*t)	z = sin(3*t)
x = sin(2*t)	y = sin(4*t)	z = sin(8*t)
x = sin(2*t)	y = sin(3*t)	z = sin(4*t)
x = sin(8*t)	y = cos(8*t)	z = (t1 - 180)/180
x = sin(4*t)	y = cos(4*t)	z = sin(4*t)*cos(8*t)
x = sin(t)*sin(4*t)	y = sin(t)*cos(4*t)	z = cos(t)
x = sin(t)*sin(4*t)	y = cos(4*t)	z = cos(t)
x = sin(t)	y = sin(2*t)	z = cos(4*t)
x = sin(4*t)	y = sin(4*t)	z = cos(5*t)
x = sin(4*t)	y = cos(4*t)	z = cos(5*t)
x = sin(2*t)*sin(6*t)	y = sin(2*t)*cos(6*t)	z = (t1 - 180)/180

Voorbeeld funkties kromme vlakken		
interval	oorsprong	funktie
2	(160,140)	z = 80*exp(-x*x-y*y)
105	(160,120)	z = 30*exp(-cos(sqr(x*x+y*y)/16))
18	(160,100)	r = sqr(x*x + y*y):z = 50*sin(r)/r
2	(160, 90)	z = 30*(x*x - y*y)
150	(160,120)	r = sqr(x*x + y*y)*[pi]/180: z = 35*(cos(r)-cos(3*r)/3+cos(5*r)/5)

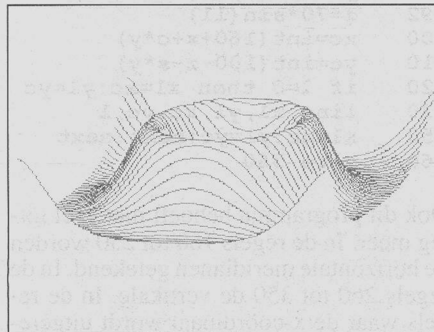
Figuur 5

De x en de y moeten we dus laten lopen in een bepaald interval, bijvoorbeeld van -2 tot 2. Hieruit kan de z-coördinaat berekend worden. Zo worden er steeds punten verkregen (x, y, z). Het programma dat ruimtelijke vlakken tekent volgt nu:

```
10      rem 3d grafiek
100     input"interval";f1
110     c=.5*cos(.25*{pi}):s=.5
        *sin(.25*{pi})
120     dim h(255):vi=0
130     for l=0 to 255
140     h(l)=1000:next
150     f=f1/115:hires 0,1
160     for yy=-115 to115 step 5
170     for xx=-115 to115 step 3
180     x=xx*f:y=yy*f
190     z=30*exp(-cos(sqr(x*x
        +y*y)/16))
200     xc=int(160+xx+c*yy)
210     yc=int(120-z-s*yy)
220     if yc>h(int(xc/3)) then
        vi=0:goto 260
230     if vi then line
        x1,y1,xc,yc,1
240     h(int(xc/3))=yc:vi=1
250     x1=xc:y1=yc
260     next:vi=0:next
270     goto 270
```

Het begin van het programma spreekt voor zich. In regel 120 en 130 wordt er een array van 255 op 1000 geïnitialiseerd. Waarvoor we dat nodig hebben wordt straks duidelijk. We laten de x en de y lopen van -115 tot 115, waardoor eerst de voorste lijnen en daarna de achterste lijnen van de figuur worden getekend. In regel 180 worden deze waarden aangepast, door ze met f te vermenigvuldigen. Zo kunt u als u wilt een kleiner of groter interval nemen dan {-115,115}. In regel 190 wordt de z-waarde berekend. Als u een klein interval heeft gekozen, dan moet de z-coördinaat wat vergroot worden, omdat er anders haast een plat vlak op het beeld verschijnt. Daarom wordt de z in regel 190 alvast met een constante vermenigvuldigd.

Nu werpt zich een probleem op. Als we bijvoorbeeld een bult in het midden van de tekening hebben, dan zullen de punten van het vlak achter de bult toch getekend worden, ondanks dat ze eigenlijk niet zichtbaar zijn. Ze worden dan gewoon dwars door de bult heen getekend. Omdat we dit niet willen, houden we in het array h, voor alle xc-waarden, de kleinste yc-waarden bij. Let op, de kleinste yc waarde is het punt dat het hoogste op het scherm staat. Als we bij een bepaalde xc een yc vinden die nog kleiner is, dan wordt het punt getekend. Omdat de figuur van voor naar achter wordt getekend, ligt dit punt lager dan een al eerder getekende lijn en is daarom onzichtbaar. Is de yc groter, dan betekent dat, dat er al een punt gevonden is dat hoger ligt. Oftewel, het punt met de betreffende yc waarde is niet zichtbaar, en moet ook niet getekend worden. In regel 220 wordt gekeken of de yc-waarde de kleinste is tot nu toe, voor de betreffende xc-waarde. Is dit niet het geval, dan wordt het punt niet getekend.



Figuur 4

Omdat we de x met step 3 laten lopen, moeten we, alvorens de yc-waarden te vergelijken, eerst de xc door 3 delen, anders komen er gaten in de grafiek. Evenals in het vorige programma wordt er steeds een lijn getrokken tussen twee opeenvolgende punten. Nu kunt u ook begrijpen waar de variabele vi voor dient. Telkens als er een punt is gevonden dat zichtbaar is, wordt vi op 1 gezet. Anders heeft vi de waarde 0. Als het volgende punt ook zichtbaar is, wordt er een lijn getrokken, anders niet. De variabele vi geeft dus steeds aan of het vorige punt zichtbaar was. Aan de hand daarvan kan dan bekeken worden of er een lijn moet worden getrokken of niet. Doordat we in de binnenste loop de x laten lopen, krijgen we steeds horizontale lijnen. De y-step in regel 160 bepaalt hoe ver deze horizontale lijnen van elkaar af liggen. In **figuur 4** kunt u het resultaat zien van het maken van een gebogen vlak. In figuur 5 staan weer wat leuke vlak-functies. Zo'n functie moet u in regel 190 plaatsen. De oorsprong moet in regels 200-210 worden ingevuld. (De x-coördinaat van de oorsprong moet op de plaats, waar nu '160' staat, ingevuld worden; de y-coördinaat op de plaats van '120'.)

Als laatste moet nog gezegd worden, dat er in het programma niet getest wordt of de xc- en yc-waarden buiten het beeld vallen. Dit wordt niet gedaan, om de snelheid te verhogen. Het kan nu dus wel voorkomen dat het programma eruit klapt met een foutmelding. U moet dan het interval kleiner maken en/of de vergrotingsfactor van de z kleiner maken. Als alleen de bovenkant niet op het scherm past, kunt u ook het middelpunt wat naar onder verplaatsen. Dit kunt u doen door de waarde 100 in de transformatie formule te vergroten.

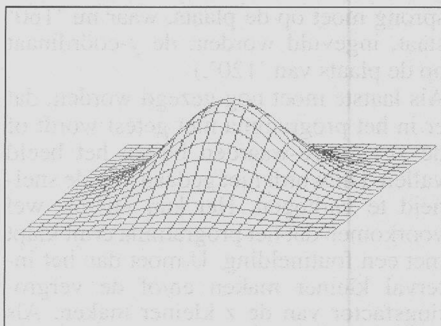
Rastervlakken

In het vorige programma hebben we steeds horizontale lijnen getrokken. We kunnen daarnaast ook vertikale lijnen trekken. Het resultaat daarvan is een rastervlak. In **figuur 6** kunt u zo'n rastervlak zien. Hieronder geven we het programma voor zo'n rastervlak. Daar dit ongeveer gelijk is aan het vorige programma, leggen we het niet uit.

```
10      rem rastervlak
100     input"interval";f1
110     c=.5*cos(.25*{pi}):s=.5
        *sin(.25*{pi})
120     dim h1(255):dim
        h2(255):vi=0
130     for l=0 to 255
140     h1(l)=1000:h2(l)=1000:
        next
```

```
150     f=f1/115:hires 0,1
160     for yy=-115 to115 step 10
170     for xx=-115 to115 step 3
180     x=xx*f:y=yy*f
190     z=80*exp(-x*x-y*y)
200     xc=int(160+xx+c*yy)
210     yc=int(140-z-s*yy)
220     if yc>h1(int(xc/3)) then
        vi=0:goto 260
230     if vi then line
        x1,y1,xc,yc,1
240     h1(int(xc/3))=yc:vi=1
250     x1=xc:y1=yc
260     next:vi=0:next
265     for xx=115 to -115 step
        -10
270     for yy=-115 to115 step 3
280     x=xx*f:y=yy*f
290     z=80*exp(-x*x-y*y)
300     xc=int(160+xx+c*yy)
310     yc=int(140-z-s*yy)
320     if yc>h2(int(xc/3)) then
        vi=0:goto 360
330     if vi then line
        x1,y1,xc,yc,1
340     h2(int(xc/3))=yc:vi=1
350     x1=xc:y1=yc
360     next:vi=0:next
370     goto 370
```

Ook bij dit programma kunt u de functies uit figuur 5 proberen. De functie moet u nu in regel 190 en 290 zetten. De oorsprong moet in regels 200-210 en 300-310 staan.



Figuur 6

Bollen, cilinders en kegels

Bollen, cilinders en kegels zijn eigenlijk niets anders dan gebogen vlakken. Toch behandelen we ze apart, omdat ze niet zo makkelijk met het vorige programma kunnen worden gemaakt. Als voorbeeld nemen we de bol. In de 3D-ruimte heeft deze normaal de volgende vergelijking:

```
x*x + y*y + z*z = r*r
```

Hierin is r de straal van de bol. Het middelpunt van de bol is de oorsprong. De bovenstaande vergelijking wordt ook wel de cartesische vergelijking van de bol genoemd. De bol kan echter ook worden beschreven door een parametervoorstelling. Voor de bol is de parametervoorstelling:

```
x = r * cos(u) * cos(t)
y = r * cos(u) * sin(t)
z = r * sin(u)
```

Algebraische voorstelling gebogen vlakken		
Gebogen vlak	**Vergelijking**	**Parametervoorstelling**
Bol middelpunt O straal r	x*x + y*y + z*z = r*r	x = r*cos(u)*cos(t) y = r*cos(u)*sin(t) z = r*sin(u)
Cilinder straal r as is z-as	x*x + y*y = r*r	x = r*cos(t) y = r*sin(t) z = u
Kegel tan vd halve tophoek is m	x*x + y*y - m*m*z*z = 0	x = m*u*cos(t) y = m*u*sin(t) z = u

Figuur 7

De r is wederom de straal van de bol. Door de u van -90 tot 90 graden te laten lopen, en de t van 0 tot 360 graden te laten lopen, wordt de hele bol beschreven. Door de u constant te houden en de t te laten lopen, worden er horizontale meridianen verkregen; door de t constant te houden en de u te laten lopen, worden er vertikale meridianen verkregen. In **figuur** 7 staan de vergelijkingen voor een bol, een cilinder en een kegel bij elkaar. We geven als laatste het programma om een bol op het scherm te tekenen. U kunt dit programma zelf aanpassen voor het tekenen van een cilinder en een kegel.

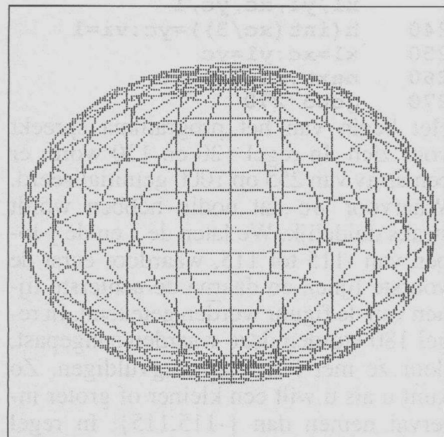
```
100     rem 3d bol
110     c=.5*cos(.5*{pi}):s=.5
        *sin(.5*{pi})
150     f=f1/115:hires 0,1
160     for l=-90 to 90 step 15
165     l1=l*{pi}/180
170     for b=0 to 360 step 3
175     b1=b*{pi}/180
190     x=1.2*70*cos(l1)*cos(b1)
191     y=70*cos(l1)*sin(b1)
192     z=70*sin(l1)
200     xc=int(160+x+c*y)
210     yc=int(100-z-s*y)
220     if b=0 then x1=xc:y1=yc
230     line x1,y1,xc,yc,1
250     x1=xc:y1=yc:next:next
260     for b=0 to 360 step 15
265     b1=b*{pi}/180
270     for l=-90 to 90 step 3
275     l1=l*{pi}/180
290     x=1.2*70*cos(l1)*cos(b1)
291     y=70*cos(l1)*sin(b1)
292     z=70*sin(l1)
300     xc=int(160+x+c*y)
310     yc=int(100-z-s*y)
320     if l=0 then x1=xc:y1=yc
330     line x1,y1,xc,yc,1
350     x1=xc:y1=yc:next:next
360     goto 360
```

Ook dit programma behoeft niet veel uitleg meer. In de regels 160 tot 250 worden de horizontale meridianen getekend. In de regels 260 tot 350 de vertikale. In de regels waar de x-coördinaat wordt uitgerekend (bijvoorbeeld regel 190), kunt u zien dat de x-coördinaat nog met een factor 1.2 wordt vermenigvuldigd. Dit hebben we gedaan, omdat een pixel op het beeld niet precies vierkant, maar een beetje rechthoekig is. Om te voorkomen dat er een ovaal op het scherm komt, hebben we de x-coördinaat met 1.2 vermenigvuldigd.

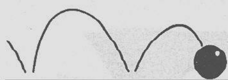
Toen we over de projectie van 3D op 2D spraken, zeiden we dat we voor hoek g tussen de y-as en de x-as 45 graden namen. In dit geval nemen we voor g 90 graden, omdat anders de bol vervormd wordt. De waarden voor de variabelen c en s veranderen dus (zie regel 110). U kunt in **figuur 8** het resultaat zien van de drie-dimensionale bol.

We hebben nu ongeveer alle variaties gezien op het vlak van de drie dimensionale ComputerKunst. In de volgende aflevering zullen we verder gaan met het onderwerp computerkunst. Daarin zullen onder andere de volgende kunstvormen de revue passeren: cirkelkunst en parket metamorfoses. En als schijfje citroen op deze cocktail behandelen we de mega scroll, waarmee zowel tekst als high resolution schermen met grote snelheid gescrolld kunnen worden.

Hylke Sprangers en Michel de Boer

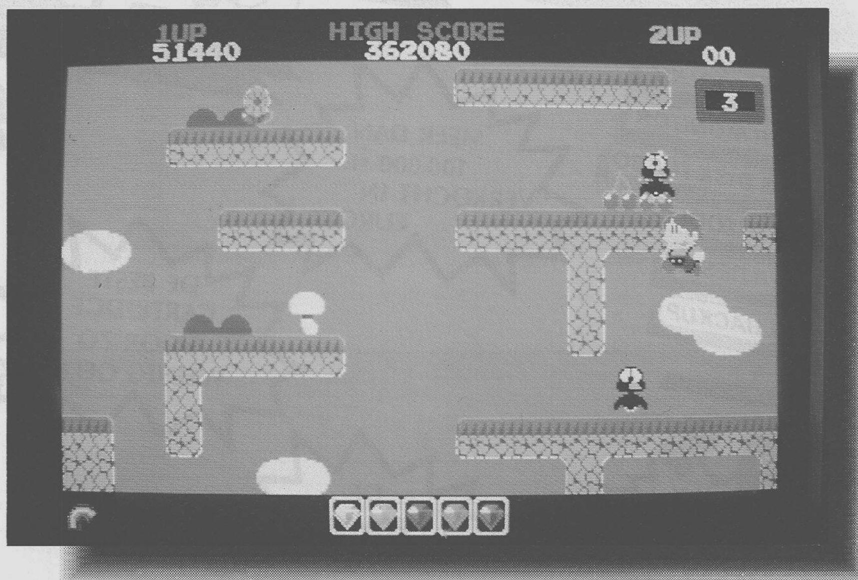


Figuur 8



Rainbow Islands en Omega

Software voor C-64 en Amiga



Rainbow Islands

Ocean komt met een nieuwe topper: Rainbow Islands. Dit programma zal zondermeer doodgeverfd worden als een opvolger van New-Zealand story. Toch is dit niet correct. De grafische afwerking is mooi om te zien en heeft jegens de kinderlijke afwerking de zelfde aantrekkingskracht als New-Zeeland Story, maar daar houdt het dan verder ook mee op.

Het is een horizontaal scrollend actie spel, waarbij men het hoogste punt moet zien te bereiken. Daar wachten hem een schatkist gevuld met fruit en het begin van een nieuw, zwaarder, level. Om de vier levels zal men een kamer tegenkomen met een monster dat een aantal keren geraakt moet worden voordat het sterft, waarna de bekende kist met fruit komt en een superfruit. Deze levert 100.000 punten op en een extra leven.
Elk van de eilanden heeft zijn eigen thema. Bijvoorbeeld: insecten, gevecht en speelgoed. Maar hoe komt men aan de naam Rainbow Islands? Simpel eigenlijk. Men bestuurd het kereltje dat zich door de eilanden heen moet banen op de bekende wijze: links, rechts en omhoog voor springen. De vuurknop echter zet voor hem een regenboog neer waarover hij heen kan lopen. Deze regenboog heeft drie functies. Als ons kereltje op het hoogste punt staat, kan hij springen naar een hoger gelegen stuk bereiken dat normaliter niet bereikbaar is of een nieuwe regenboog maken en zo een trapeffect creeëren. De tweede functie is het opsluiten van een vijand. Als men namelijk de tegenstander insluit in een regenboog kan hij je niet volgen of kwaad doen.
De derde functie is de vijand te raken met het einde van de regenboog. De vijand wordt dan in voedsel of een extra veranderd hetgeen punten of krachten oplevert. Behalve eetbare goederen zijn er ook de volgende nuttige artikelen te vinden. Flesjes, deze zijn er in twee kleuren. De rode geven een extra boog aan de regenboog tot een maximum van 3 en de gele geven de regenboog snelheid zodat ze veel sneller gemaakt worden. Tevens kan men sportschoenen tegenkomen waarmee men dan sneller kan lopen. Het spel doet niet onder voor New-Zealand Story, sterker nog: ik vond hem zelfs in menig opzicht beter.

Omega

Ieder begin van een recensie is moeilijk. Zeker als je over een spel schrijft met gemengde gevoelens. Maar doch ik beschouw dit spel als dermate uniek dat ik toch door de figuurlijk zure appel moet bijten.

Wat is Omega? Omega is een programma wat meer een studiepakket is dan een spel, maar dan in een aangenaam pakje gekleed. In dit spel is het zaak een tank te ontwikkelen met een bepaald budget. Dit ontwikkelen bestaat uit twee fases. De eerste is het kiezen van de noodzakelijke hardware en de tweede behelst de kunstmatige intelligentie (KI) van de tank. De eerste fase wordt beperkt door een gebrek aan budget. Je krijgt in eerste instantie maar 1000 credits. Dus kom je niet verder dan standaard apparatuur met 1 scanner. Het ontwikkelen van de KI is het waar het voornamelijk om draait. Hierin schrijf je een programma, waarin de handelingen omschreven worden, die de tank moet ondernemen. De taal die hiervoor gebruikt wordt is niet uitgebreid, maar wel zwaar gestructureerd. Dit is voor een beginneling dus een goede manier om iets van gestructureerd programmeren te leren en na te gaan wat het gevolg daarvan is. Indien dit gedaan is, kan men de tank op een terrein testen en eventueel verbeteren. Als men uiteindelijk de tank test en hij wordt als succesvol gezien, zal men promotie maken een meer geld krijgen zodat men meer apparatuur kan plaatsen. Het testen gebeurt dus op een terrein naar keuze. Het spel wordt geleverd met drie terreinen, maar het is mogelijk om een eigen terrein te ontwikkelen. Het beeld is over de gehele lijn goed afgewerkt. Maar de terreinen waar de gevechten plaatsvinden, zijn 'basic' qua afwerking, doch zeker niet slecht. Het spel heeft in mijn ogen een duidelijke aantrekkingskracht voor de speler, die ervan houd om de nodige logische (programmeer)-problemen op te lossen en zijn vaardigheid wil testen tegen dat wat de computer hem voorschotelt.
Persoonlijk vond ik het oude programma 'Chipwits' van Epyx aangenamer voor wat betreft het grafische gedeelte. Maar de optie om de KI te ontwikkelen is hier aanzienlijk beter benut.

Lawrence van Rijn



Amiga Games

Wie wil dat nou niet? Rijden in een Ferrari F40 dat je leven er van af hangt, de politie op je hielen en zorgen dat je ze voor blijft. Dat is de opdracht bij het spel F40 Pursuit Simulator; wilde achtervolgingen door de Verenigde Staten.

Terwijl je je doel probeert te bereiken race je door vier staten en krijg je met allerlei moeilijkheden te doen. De Ferrari haalt zo'n 250 kilometer per uur wat hard genoeg is om de politiewagen ver achter je te laten.
Wegomleggingen zijn hindernissen waarvoor je een oplossing dient te zoeken. De route volg je door middel van vier landkaarten van de staten waar je doorheen crossed. Maar denk niet dat de korste routes ook altijd de beste zijn. Want voordat je het weet, doemen ineens blauwe zwaailichten op en de gillen de sirenes om je heen.
Terwijl je bezig bent, realiseer je vaak niet eens hoe snel de tijd gaat, tenminste op het scherm. Voordat je er erg in hebt wordt het nacht, waardoor het rijden er niet makkelijker op wordt.
De F40 is voorzien van de nieuwste radar, die zonder problemen de politie in de gaten houdt en waar zij deze keer weer een blokkade hebben ingericht. Dat item doet een beetje denken aan Test Drive. De 'road blocks', zoals ze in het Engels heten, zijn er speciaal om ervoor te zorgen, dat je van richting veranderd.
Met plotselinge versperringen weet de F40 zich wel raad: de auto kan om zijn as draaien tot 360 graden. Dus als er opeens iets aan de knikker mocht zijn, kan de bestuurder in één keer omdraaien.
Het spel heeft prachtige graphics en daardoor realistische 3D. Maker is Titus Software.



F40 Persuit Simulator

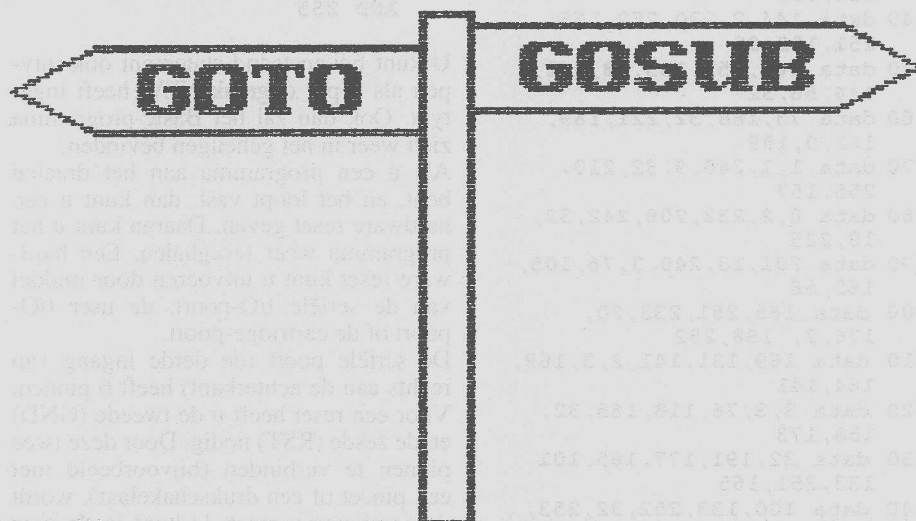
Altijd 'in'

Accolade brengt vier verschillende spellen uit voor de prijs van één: **All-time Favourites**. Test Drive is inmiddels overbekend en bij veel Amiga-fans geliefd. Terwijl je in een super-bolide over de wegen raast, maakt, net als bij F40 Pursuit Simulator, de politie je het een beetje moeilijker. Zorg in ieder geval dat je op tijd bij de benzinepompen staat, anders komt je zonder te staan en het spel is, zonder één crash en politie, afgelopen.
Mean 18, ook op deze compilatie te vinden, is een programma waar de speler zich op een golfbaan bevindt. Mean 18 maakt van een amateur een heuze prof.
Speel golf op een beroemde baan in de Famous Courses Volumes I & II. Wie de beroemde banen van Pebble Beach en St. Andrews achter de rug heeft, kan zich gaan wagen aan het zelf ontwerpen van banen naar keuze.
Hardball is het vierde spel op deze disk. Volgens maker Accolade is dit spel nog echter dan basketbal op televisie. Het is alsof je zelf meespeelt.
Accolade All-Time Favourites komt volgende maand uit.

Plague

UGA uit Zeist stuurde ons een pre-view van het spel '**Plague**'. Waarom Plague (pest) nu juist deze naam heeft, is niet echt duidelijk. Het is het eerste Nederlandse spel dat verkocht is aan een professioneel bedrijf: Innerprise (van onder andere Lost Dutchman Mine). Deze preview doet qua naam zichzelf (gelukkig) geen eer aan. Het is een science-fiction-achtig spel, waarbij de maker erg veel fantasie heeft gebruikt. De speler is een vechtersbaas die zich door allerlei levels moet vechten, omdat er overal gevaar op de loer ligt in de vorm van beesten en 'future'-figuren, die het de speler erg moeilijk kunnen maken.
Of het spel moeilijk te spelen is, viel niet vast te stellen, omdat we met een demoversie te maken hadden. Het ziet er in ieder geval niet slecht uit voor spel van vaderlandse gronden. De grafiek is van professionele kwaliteit en het beeldscherm wordt bijna tot aan de randen gevuld.

Tips & Trucs 64



Eindelijk is het dan zover. De kwaadaardige koning winter heeft het moeten afleggen tegen de charmes van de schone prinses lente. Een nieuwe lente en een nieuw geluid: de nieuwe Commodore Info is weer uit. In deze aflevering van Tips & Trucs bieden wij onder andere de volgende onderwerpen: functietoetsen, alles over reset, goto en gosub met variabelen en de BRK-instructie.

Lezers worden overigens van harte uitgenodigd om eigen tips & trucs in te sturen. Ook ideeën voor deze rubriek zijn van harte welkom. Stuur alles naar: Commodore Info, Postbus 43048, 1009 ZA Amsterdam.

Auto Number

Het programma Auto Number, dat we hieronder geven, is een hulpprogramma dat u kunt gebruiken als u zelf Basic-programma's schrijft. Het programma genereert telkens regelnummers. Achter elk regelnummer kunt u gewoon een regelcode ingeven. Als u daarna op return drukt, dan wordt het volgende regelnummer gegenereerd. Zo krijgt u een programma dat netjes genummerd is.

We zullen hieronder uitleggen hoe u het programma Auto Number moet gebruiken. Als u het programma heeft ingetypt (en gesaved), moet u RUN intypen. Vervolgens kunt u het programma weggooien met NEW. Het machinetaal programma, dat automatisch regelnummers genereert, zit nu in het geheugen vanaf adres 49152.

Auto Number kan nu als volgt gestart worden:

```
SYS 49152, STARTREGEL, STEP
```

Voor STARTREGEL moet u het regelnummer ingeven, waarmee u wilt beginnen. Dit regelnummer wordt steeds verhoogd met de opgegeven STEP. Als u bijvoorbeeld het volgende ingeeft: sys 49152,100,10 dan begint u met regel 100. De volgende regel wordt dan 110, enzovoort.

Als u wilt stoppen met programmeren, dan moet u gewoon een return ingeven achter de regel. Het programma zal dan automatisch stoppen. U kunt daarna wel gewoon weer verder gaan met het regelnummer waar u gebleven was. U moet dan intypen:

```
SYS 49160
```

U kunt natuurlijk ook gewoon weer opnieuw starten met sys 49152, startregel, step. Het programma dat al in het geheugen staat, blijft dan gewoon staan.

```
5  rem auto number
10 for x=49152 to 49290
20 read a:poke x,a:i=i+a:next
30 if i=18127 then
   print"ok":end
40 print"fout in data"
```

Tabel funktie toetsen		
Funktie toets	ASCII Waarde	Scan waarde adres 197
F1	133	4
F2	134	
F3	135	5
F4	136	
F5	137	6
F6	138	
F7	139	3
F8	140	

Figuur 1



```
400 data 32,253,174,234,32,
     101,192,234
410 data 169,19,141,2,3,169,
     192,141
420 data 3,3,96,32,25,192,
     76,134
430 data 164,24,169,9,101,251,
     133,251
440 data 144,2,230,252,165,
     251,133,99
450 data 165,252,133,98,162,
     144,56,32
460 data 73,188,32,221,189,
     162,0,189
470 data 1,1,240,9,32,210,
     255,157
480 data 0,2,232,208,242,32,
     18,225
490 data 201,13,240,3,76,105,
     165,56
500 data 165,251,233,20,
     176,2, 198,252
510 data 169,131,141,2,3,169,
     164,141
520 data 3,3,76,118,165,32,
     158,173
530 data 32,191,177,165,101,
     133,251,165
540 data 100,133,252,32,253,
     174,32,158
550 data 183,142,27,192,56,
     165,251,237
560 data 27,192,133,251,165,
     252,233,0
570 data 133,252,96
```

De werking van het bovenstaande programma kunnen we ook simuleren in Basic. Hier komt het Basic-programma dat automatisch regelnummers genereert:

```
5 rem auto number Basic
6 a=100
7 print"{CLR}"a;:poke
   19,1:input a$:poke
   19,0:print
8 ?"a="a"+10":goto7":poke
   198,3:poke 631,19:poke
   632,13:poke 633,13:end
```

De variabele a in regel 6 is het regelnummer waarmee begonnen wordt. Door in adres 19 een 1 te zetten, wordt het toetsenbord het huidige I/O-apparaat. Dit heeft als consequentie dat het vraagteken bij de INPUT in regel 7, niet op het beeldscherm wordt gezet.

Reset

Bij een reset wordt de computer weer in de beginstand gezet. U krijgt dan gewoon weer het beginbeeld, wat u ook ziet als u de computer aan doet. Het verschil is echter, dat het geheugen van de computer er na een reset nog net zo uit ziet als voor de reset. Dit heeft als voordeel dat u eventuele machinetaal programma's die in het geheugen staan, nog gewoon kunt uitvoeren. Bij Basic-programma's ligt de zaak wat moeilijker. De Basic-pointers zijn nu immers ook weer teruggezet, alsof er zich geen programma in het geheugen bevindt. We kunnen het programma echter nog wel terughalen, en wel met het volgende statement:

```
POKE 2050,8:SYS 42291:POKE
    46,PEEK(35)-PEEK(781)+2
    AND 255
```

U kunt bovenstaand statement ook intypen als u per ongeluk NEW heeft ingetypt. Ook dan zal het Basic-programma zich weer in het geheugen bevinden.
Als u een programma aan het draaien bent, en het loopt vast, dan kunt u een hardware reset geven. Daarna kunt u het programma weer terughalen. Een hardware reset kunt u uitvoeren door middel van de seriële I/O-poort, de user I/O-poort of de cartridge-poort.
De seriële poort (de derde ingang van rechts aan de achterkant) heeft 6 pinnen. Voor een reset heeft u de tweede (GND) en de zesde (RST) nodig. Door deze twee pinnen te verbinden (bijvoorbeeld met een pincet of een drukschakelaar), wordt de computer gereset. U kunt in de user manual de pinverdeling vinden van de seriële poort.
De user-poort zit helemaal aan de rechterkant van de computer. Van de 24 pinnen die de user-poort heeft, moet u de eerste (GND) en de derde (RST) pin verbinden. Ook nu kunt u in de user manual de pinverdeling vinden.
U kunt echter ook een software reset genereren. Dit kunt u doen met:

```
SYS 64738
```

Dit heeft dezelfde uitwerking als een hardware reset. Er is ook een aantal minder drastische resets, zoals sys 64764. In dit geval zullen niet alle registers en dergelijke weer opnieuw geset worden. In principe heeft u aan zo'n sys dus niet zoveel (al gaat hij wel een stuk sneller).

Er is ook een manier om een reset uit te voeren, zonder dat alle data verloren gaan. Het voordeel hiervan is, dat je de variabelen nog kunt bekijken, om bijvoorbeeld uit te zoeken wat er precies mis ging in een programma. We zullen een voorbeeld geven. Stel u zit in de hires mode. Nu breekt het programma af met een error. Nu kunt u de reset ingeven. De computer zal dan weer over gaan tot het normale lores scherm. Nu kunt u de variabelen checken. Hier komt de reset:

```
POKE 648,4:SYS 64789:SYS
    58451:SYS 58784
```

Functietoetsen

De Commodore 64 kent 4 functietoetsen, aan de rechterkant van het toetsenbord. Samen met de shift-toets kunnen we zo 8 combinaties maken (F1 t/m F8). Vaak wordt er niet zoveel met deze toetsen gewerkt. Toch kunnen ze best handig zijn. Vooral in interactieve programma's (dit zijn programma's waarin de gebruiker rechtstreeks invloed kan uitoefenen op de loop van het programma, bijvoorbeeld door keuzemenu's), kan veel met de functietoetsen gedaan worden, ten einde het programma gebruikersvriendelijk te maken. Er kan op een aantal manieren getest worden of een functietoets ingedrukt is.
Ten eerste hebben de functietoetsen een ASCII-waarde. Zo heeft F1 de waarde 133. In **figuur 1** staan de ASCII-waarden van de functietoetsen. In een programma kan nu eenvoudig, met behulp van de CHR$-functie, getest worden of er een functietoets is ingedrukt. Het onderstaande programma geeft hier een voorbeeld van:

```
5 rem functie toetsen 1
10 get a$: if a$="" then 10
20 if a$=chr$(133) then
   print"f1"
30 if a$=chr$(134) then
   print"f2"
40 if a$=chr$(135) then
   print"f3"
50 if a$=chr$(136) then
   print"f4"
60 goto 10
```

Er is nog een manier om te testen of een functietoets is ingedrukt. In adres 197 wordt de laatste toets bijgehouden die is ingedrukt. In figuur 1 staan de waarden van de functietoetsen voor adres 197. Zo heeft F1 de waarde 4. Ook hier geven we een kort voorbeeldprogramma.

```
5 rem functie toetsen 2
10 a=peek(197)
20 if a=4 then print"f1"
30 if a=5 then print"f3"
40 if a=6 then print"f5"
50 if a=3 then print"f7"
60 goto 10
```

U ziet dat de combinaties van de vier functietoetsen met de shift-toets niet in de tabel voorkomen. Dit komt omdat adres 197 geen toetsen bijhoudt in combinatie met de shift.

Als laatste geven we een machinetaal programma om een indruk te geven hoe de functietoetsen in de praktijk gebruikt kunnen worden. Het programma moet gestart worden met sys 49152. Door nu een functietoets in te drukken wordt de computer vertraagd. Als F7 wordt ingedrukt, wordt de computer helemaal even gestopt met het werk wat hij aan het doen was

(bijvoorbeeld LIST of RUN). Dit gaat door tot F7 weer los wordt gelaten. Dit programma kan handig zijn bij het listen van een programma. U weet dat het listen vertraagd wordt door de CTRL-toets in te drukken. Wel, met dit programma heeft u er nog een aantal vertragingstoetsen bij, elk met een andere vertragingsfactor. De functietoetsen vertragen ook bij het runnen van een programma. Hier komt het programma.

```
5 rem functie toetsen
10 for x=49152 to 49218
20 read a:poke x,a:i=i+a:next
30 if i=9899 then sys
    49152:end
40 print"fout in data"
400 data 120,169,13,141,20,3,
    169,192
410 data 141,21,3,88,96,162,
    0,160
420 data 0,165,197,201,4,208,
    10,232
430 data 208,253,200,192,48,
    208,248,160
440 data 0,201,5,208,6,232,
    208,253
450 data 200,208,250,201,6,
    208,8,232
460 data 234,234,208,251,200,
    208,248,32
470 data 159,255,165,197,201,
    3,240,247
480 data 76,49,234
```

Uitgebreide GOTO en GOSUB

Het standaard Basic van de Commodore 64 staat niet toe dat met de **GOTO**-of **GOSUB**-instructie naar de waarde van een variabele wordt gesprongen. Zo zal het volgende programma de foutmelding: 'UNDEF'D STATEMENT ERROR' geven.

```
10 a=100
20 gosub a
...
100 rem subroutine
...
```

Het argument bij de instructies GOTO en GOSUB moet een constant getal zijn en geen variabele, zoals in het bovenstaande programma. Soms is het echter zeer handig om naar de waarde van een variabele te springen in plaats van naar een constant getal. (Om bijvoorbeeld omvangrijke ON-GOTO-constructies te vermijden.)
Door een kleine verandering in de ROM-routine voor de GOTO- en GOSUB-instructies aan te brengen, is het mogelijk om zowel naar constante getallen als naar variabelen te springen. Hieronder volgt het programma dat deze verandering aanbrengt.

```
5 rem goto en gosub met
    variabelen
10 for i=40960 to 49151
20 poke i,peek(i):next
30 for i=48979 to 48984
40 read a:poke i,a:next
50 for i=43168 to 43170
60 read a:poke i,a:next
70 poke 1,peek(1) and not 1
80 data 32,138,173,76,247,183
90 data 32,83,191
```

Nadat dit programma is uitgevoerd, kan met de GOTO- en GOSUB-instructies naar variabelen worden gesprongen. U kunt dit programma bijvoorbeeld als subroutine in uw eigen programma opnemen. Deze subroutine moet dan als eerste uitgevoerd worden, zodat in de rest van het programma gebruik kan worden gemaakt van de uitgebreide GOTO- en GOSUB-instructies.
Om de werking van dit programma te begrijpen, is enige kennis van de Basic-interpretator nodig. De uitleg van het programma valt daarom buiten het bestek van dit artikel.

Schermmanipulaties

Met de Basic PRINT-instructie kunnen alle mogelijke schermmanipulaties, zoals het afdrukken van tekst, wissen van het scherm en plaatsen van de cursor, uitgevoerd worden. Door gebruik te maken van een aantal ROM-routines kunnen deze schermmanipulaties ook vanuit machinetaal programma's uitgevoerd worden; het is dus niet nodig om zelf routines voor deze schermmanipulaties te schrijven. In deze aflevering zullen we de ROM-routines voor schermmanipulaties behandelen. Bij elke routine geven we het hexadecimale startadres met daaronder de uitleg van de routine.

$FFD2:

Deze routine print een karakter op het scherm. De ASCII-waarde van dit karakter moet voor de aanroep van de routine in de accumulator worden gezet. Dus: 'LDA #"A":JSR $FFD2' is equivalent met 'PRINT"A";'. Let op de punt-komma bij de PRINT-instructie; de cursor wordt na het printen van een karakter niet op het begin van de volgende regel gezet. Als we willen dat de cursor wel op de volgende regel gezet wordt, moeten we nog de ASCII-waarde 13 'printen'. (LDA #13:JSR $FFD2).

In principe kunnen met deze ROM-routines alle schermmanipulaties worden uitgevoerd. Om bijvoorbeeld het scherm te wissen moet ASCII-waarde 147 worden geprint en om een string te printen moet elke letter van de string afzonderlijk worden afgedrukt. Er is echter een aantal ROM-routines die deze schermmanipulaties uitvoert. Deze routines zijn de volgende.

$AB1E:

Met deze routine kan een hele string in plaats van een enkel karakter worden uitgeprint. De string moet als een rij ASCII-waarden afgesloten door een 0 in het geheugen staan. De accumulator en het y-register moeten respectievelijk het lo- en hi-byte van het beginadres van deze rij bevatten. Het volgende programma verduidelijkt het gebruik van deze routine.

```
       lda #<string
       ldy #>string
       jsr $ab1e
       rts
string .asc "daar gaat ze"
       .byt 0
```

De cursor wordt na het printen van de string niet op de volgende regel gezet. Het programma is dus equivalent met 'PRINT"daar gaat ze". Om de cursor wel op de volgende regel te laten zetten, moet als laatste waarde nog de waarde 13 worden opgenomen in de string. De laatste regel van het programma moet dan veranderd worden in '.byt 13,0'.

$E544:

Deze routine wist het scherm. JSR $E544 is dus equivalent met 'PRINT"{SHIFT-CLR}";'.

$E566:

Deze routine zet de cursor linksboven op het scherm. Dit is equivalent met 'PRINT"{HOME}";'.

$E87C:

Deze routine zet de cursor op de volgende regel. Als de cursor op de onderste regel stond, dan wordt het scherm omhoog gescrolld.

$E8EA:

Deze routine scrollt het scherm een regel omhoog. De cursor blijft op dezelfde positie staan.

$E9FF:

Deze routine wist een regel van het scherm. Voor de aanroep moet het nummer van deze regel in het x-register worden geplaatst. De regels zijn daarbij van 0 tot 24 genummerd.

$FFF0:

Met deze routine kan afhankelijk van de waarde van carry flag de cursor geplaatst worden of de positie van de cursor gelezen worden. Als de carry flag uit staat, kan de cursor geplaatst worden. Het x- en



het y-register moeten dan respectievelijk het regel- en kolomnummer van de positie bevatten voor de aanroep van de routine. Als de carry flag aan staat, kan de positie van de cursor gelezen worden. Het x- en het y-register bevatten dan na de uitvoering van de routine het regel- en kolomnummer van de positie.

$FFED:

Deze routine retourneert het aantal kolommen op het scherm in het x-register en het aantal regels in het y-register.

Het printen van getallen vanuit machinetaal is een vervelend karwei. Met behulp van ROM-routines kunnen twee soorten getallen worden geprint: gehele getallen tussen 0 en 65535 en twee-complement getallen tussen 32768 en 32767.
Het printen van gehele getallen tussen 0 en 65535 is het eenvoudigst. Dit kan met de volgende ROM-routine gedaan worden:

$BDCD:

Deze routine print een getal tussen 0 en 65535. Het lo-byte van het getal moet voor de aanroep in het x-register worden geplaatst en het hi-byte in het y-register.
Voor het printen van twee-complement getallen tussen -32768 en 32767 zijn drie ROM-routines nodig. Het lo-byte van het getal moet in het y-register worden geplaatst en het hi-byte in de accumulator. Daarna moeten achtereenvolgens de routines $B391, $BDDD en $AB1E worden aangeroepen. Het volgende programma print het getal 4242 op het scherm.

```
ldy #<4242
lda #>4242
jsr $b391 ;vertaal getal naar
jsr $bddd ;string
jsr $able ;print string
rts
```

Het programma vertaalt het getal eerst naar floating point-formaat en zet het floating point-formaat vervolgens om in een string. Met de routine op adres $AB1E wordt de string tenslotte uitgeprint. De twee vertalingen zijn nogal traag. Als u dus een snel programma wilt schrijven, zult u zelf routines voor het printen van getallen moeten maken. Ook voor getallen die kleiner dan -32768 of groter dan 65535 zijn, moet u zelf een routine maken. Hiervoor zijn helaas geen ROM-routines aanwezig.

BRK

Zodra de computer een BRK-instructie tegenkomt bij het uitvoeren van een machinetaal programma, wordt de uitvoering gestaakt en worden de VIC, SID-en CIA-chips gereset en wordt er een zogenaamde 'Basic warm start' uitgevoerd. Deze Basic warm start sluit eventueel geopende files, initialiseert de standaard in- en uitvoer randapparaten, leegt de stack en keert terug naar Basic. De BRK-instructie heeft precies hetzelfde effect als het tegelijkertijd aanslaan van de STOP-en RESTORE-toetsen.

Als u echter wel eens met een machinetaal monitor heeft gewerkt, dan zult u wel gemerkt hebben dat het mogelijk is om het effect van de BRK-instructie te veranderen. In plaats van het resetten van alle chips wordt nu de monitor opgestart en worden er verscheidene gegevens afgedrukt, zoals de waarde van de verschillende registers en de program counter. Deze gegevens kunnen van groot nut zijn bij het debuggen van een programma. We zullen in deze aflevering uitleggen hoe het effect van BRK-instructie veranderd kan worden.

Wat gebeurt er precies als de computer een BRK-instructie tegenkomt? Eerst wordt de program counter (PC) met één verhoogd, zodat deze naar het adres, volgend op de BRK-instructie, wijst. Dit adres wordt vervolgens op de stack gezet. Daarna wordt de BRK-flag (B) gezet. Deze flag is bit 4 van het status register (SR). Het status register wordt ook op de stack gezet. Tot slot genereert de computer een IRQ-interrupt. Dit is de interrupt die normaal 60 keer per seconde wordt uitgevoerd om het toetsenbord te scannen en de klok bij te werken. Het eerste wat deze interrupt doet is het op de stack plaatsen van de A- X- en Y- registers. Daarna wordt de BRK-flag getest. Als deze flag niet gezet is, wordt de normale interrupt-routine uitgevoerd. Is deze flag echter wel gezet, dan wordt indirect naar de vector op adres $0316 gesprongen (JMP ($0316)). Standaard wijst deze vector naar de routine op adres $FE66. Deze routine reset alle chips en voert de 'warm start' uit. Het feit dat adres $0316 een RAM-adres is, stelt ons in staat om de vector te veranderen. Door deze vector naar een zelf gemaakte machinetaal routine te laten wijzen, kunnen we het effect van de BRK-instructie wijzigen. Het onderstaande programma geeft hier een voorbeeld van. In plaats van het resetten van alle chips, wordt de program counter afgedrukt. Vervolgens keert het programma terug naar Basic door middel van de 'warm start'.

```
Labels  Mnemonics      commentaar
lda     #<break        ;wijzig de
sta     $0316          ;BRK vector
lda     #>break
sta     $0317
brk     break  tsx
lda     $0104,x
eor     #$10           ;reset B flag
sta     $0104,x
pla                    ;Y-reg van
                        stack
pla                    ;X-reg van
                        stack
pla                    ;Accu van
                        stack
plp                    ;SR van stack
lda     #<mess
ldy     #>mess
jsr     $able
pla                    ;PC van stack
tax
pla
jsr        $bdcd       ;PC afdrukken
jsr        $e37b       ;Basic warm
                        start
mess       .asc "program counter:"
.byt 0
```

In het programma wordt niets met de waarden van de A- ,X- en Y- registers gedaan. Deze worden alleen van de stack gehaald. U zou het programma zodanig kunnen aanpassen dat, naast de program counter, ook deze waarden worden afgedrukt.
Greetings from little red riding hood and the seven thumbs.

Michel de Boer & Hylke Sprangers

GEOS machinetaal (10)

Macro's, headers en GeoLinker

Deze tiende aflevering van de machinetaalcursus bevat het tweede en laatste gedeelte van het totaaloverzicht van de mogelijkheden van het GeoProgrammer-pakket. In deel 9 ben ik ingegaan op de 'features' van de GeoAssembler. Deze keer komt de rest van het verhaal over de GeoAssembler aan bod (macro's en headers), en is er aandacht voor de Geolinker.

Macro's

Eenvoudig bekeken is een macro gewoon een speciale vorm van tekstvervanging in een assembly listing. Als u een stuk machinecode hebt (dat een bepaalde taak uitvoert) dan kunt u dit stuk een naam geven. Dit een-naam-geven wordt 'macro-definition' genoemd. Een bekend macro uit de GEOS-assembly code is 'LoadW adres,value', en behoort tot de standaard macro's die in de file geosMac staan. Wat daar staat is het volgende:

```
.macro LoadW dest,value
    lda #](value)
    sta dest+1
    lda #[(value)
    sta dest+0
 .endm
```

Macro versus routine

Overal waar de naam LoadW aangeroepen wordt in uw programma ('macro invocation'), wordt het stukje code tussen de bovenste en de onderste regel van het bovenstaande voorbeeld ingevoegd. De GeoAssembler vervangt dus de tekst LoadW door die vier regels machinecode ('macro expanding'). Het aanroepen van een macro heeft wel iets weg van het aanroepen van een routine: in beide gevallen duidt een woord op het uitvoeren van een taak. Het verschil tussen een routine en een macro is, dat de code die nodig is voor het uitvoeren van die taak bij een procedure maar op 1 plaats bestaat. De procedure-aanroep betekent dan dat de computer daarheen springt. Bij een macro-aanroep wordt er helemaal niets gesprongen, het stukje code wordt op de aanroep-plaats ingevuld. Dit betekent dan ook dat er net zoveel stukjes code zijn als macro-aanroepen, terwijl er in het geval van een procedure zoals gezegd maar 1 stukje code is. Het is duidelijk dat het doorgaans niet zo slim is een ontzettend groot macro te maken, dit kan dan beter met een procedure. Macro's zijn vooral handig voor het leesbaar maken van machinetaallistings, en worden steeds toegepast voor het verrichten van kleine taken.

geosMac macro's

macro	parameters	betekenis
LoadB	dest,value	; laad dest met waarde value
LoadW	dest,value	; laad dest met [(value), dest+1 met](value)
MoveB	source,dest	; laad dest met waarde van source
MoveW	source,dest	; laad dest en dest+1 met resp. de waarden
		van source en source+1
add	source	; accu = accu + waarde van source
AddB	source,dest	; waarde van dest = w.v. source + w.v. dest
AddW	source,dest	; als AddB maar dan met words
AddVB	value,dest	; waarde van dest = waarde van dest + value
AddVW	value,dest	; als AddVB maar dan met words
sub	source	; accu = accu - waarde van source
SubB	source,dest	; trek de waarde van source van dest af
SubW	source,dest	; als SubW maar dan met words
CmpB	source,dest	; vergelijkt de waarden van source en dest
CmpBI	source,immed	; vergelijkt de waarde van source met ≠immed
CmpW	source,dest	; als CmpB maar dan met words
CmpWI	source,immed	; als CmpWI maar dan met words
PushB	source	; zet de waarde van source op de stack
PushW	source	; zet de waarden van source en source+1 op stack
PopB	dest	; haal de waarde van dest van de stack
PopW	dest	; haal de waarden van dest en dest+1 van stack
bra	addr	; zelfde als jmp addr (branch always)
smb	bitNumber,dest	; zet bit bitNumber in dest aan
smbf	bitNumber,dest	; als smb, maar de accu-inhoud wordt weggegooid
rmb	bitNumber,dest	; zet alle bits van dest behalve bit
		; bitNumber op 0
rmbf	bitNumber,dest	; als rmb, maar de accu-waarde wordt weggegooid
bbs	bitNumber,source,adress	; spring naar adres als bit
		; bitNumber in source aanstaat.
		; accu en conditiecodes ongewijzigd
bbsf	bitNumber,source,adress	; als bbs, maar accu en
		; conditiecodes worden overschreven.
bbr	bitNumber,source,adress	; spring naar adres als bit
		; bitNumber in source uitstaat.
		; accu en conditiecodes ongewijzigd
bbrf	bitNumber,source,adress	; als bbr, maar accu en
		; conditiecodes worden overschreven.

Illustratie 1

In het voorbeeldje van daarnet stond de definitie van het macro 'LoadW'. LoadW is een functie die een woord (2 opeenvolgende bytes in het geheugen) een bepaalde waarde geeft. Dat gebeurt ook inderdaad: de highbyte (een ']' betekent highbyte, zie aflevering 9) van de waarde wordt in dest+1 gezet, en de lowbyte in adres dest+0. Eigenlijk heel makkelijk.

Macro-definitie

Het definiëren van een macro kan op elke plaats in de machinetaal- listing gebeuren (behalve binnen de definitie van een an-



der macro). De macrodefinitie begint altijd met '.macro '. Daarna volgt de naam en daarna eventuele parameters (maximaal 6), gescheiden door komma's. Vervolgens komt op de volgende regel de 'echte' definitie: de code zelf. Tenslotte volgt een '.endm'- commando. De code tussen de '.macro' en de '.endm' regels is gewone GEOS-assembly, zoals u die van de vorige aflevering kent. De enige beperkingen zijn dus dat binnen een macro geen andere macro's mogen worden gedefinieerd, en dat er geen files mogen worden geïmporteerd met '.include'. Aanroepen van andere macro's in de macro-definitie is toegestaan. Indirecte recursie (macro A roept B aan en macro B roept A aan) of directe recursie (macro A roept zichzelf aan) is toegestaan, maar dit kan maar tot drie niveau's diep.

```
;voorbeeldje recursie
.macro do_something keer
    .if (keer=3)
        lda #1
        sta $1000
        do_something 2
    .elif (keer=2)
        lda #2
        sta $1001
        do_something 1
    .else
        lda #3
        sta $1002
    .endif
.endm
```

geosMac

Zoals al vermeld werd is 'LoadW' een standaard GEOS-macro. Deze standaard macro's staan in de file 'geosMac'. In mijn listings gebruik ik eigenlijk alleen maar 'LoadB' en 'LoadW', maar in geosMac staan nog veel meer handige macro's. Illustratie 1 geeft een overzicht van alle standaard macro's en hun werking. Macro's kunt u ook voor hele andere dingen gebruiken. Bijvoorbeeld voor het maken van Nederlandse of duidelijkere mnemonics:

```
.macro laad_accu waarde
    lda #waarde
.endm
.macro ga_gelijk adres
    beq adres
.endm
enzovoort.
```

Labels die in macro's gedefinieerd worden, zet de GeosAssembler om in lokale labels. Als nummers voor die lokale labels neemt hij 9999$ en naar beneden. Er is dus een gevaartje dat er een dubbele-label definitie optreedt, als de programmeur zelf al zo'n hoog label in gebruik heeft. Door nooit vier-cijferige labels beginnend met een '9' te gebruiken omzeilt u normaliter dit probleem wel. Er is een uitzondering voor de regel dat labels in een macro worden omgezet in lokale labels: als de labels een meegegeven parameter zijn gebeurt dit niet. Zie het volgende voorbeeld:

```
.macro dubbelle_lp,
 label_1, label_2, xvalue,
yvalue
         ldx #xvalue
label_2:
         ldy #yvalue
label_1:
.endm
```

Met de code:

```
dubbelle_lp binnenste,
 buitenste, #20, #255
         sta ($fd),y
         dey
   bne binnenste
   inc $fe
   dex
bne buitenste
```

Heeft tot gevolg:

```
ldx #20
buitenste:
   ldy #255
   binnenste:
      sta ($fd),y
      dey
   bne binnenste
   inc $fe
   dex
bne buitenste
```

Headers

Een GEOS-header is een diskblok van 256 bytes, waarin informatie staat over de file waaraan hij toebehoort. Iedere GEOS-file heeft 1 header. In de header staat informatie over het begin-, eind- en startadres van het programma. Bovendien staat het icon van het programma hier opgeslagen. De GEOS-fileheader is al regelmatig in mijn eerdere artikelen opgedoken. Voor het gemakkelijk construeren van een header (ieder GEOS-machinetaalprogramma heeft een header nodig, dat verwacht de GeoLinker) kent de GeoAssembler de commando's '.header' en '.endh'. Het eerste commando zorgt ervoor dat er een blok van 256 bytes gemaakt wordt. De code die na 'header' stond, wordt daarin gezet. Het tweede commando, '.endh' dient om de header te beëindigen. Als de GeoAssembler '.endh' tegenkomt terwijl er nog minder dan 256 bytes aan code na de '.header' stond wordt de rest standaard met nullen aangevuld tot 256 bytes. Illustratie 2 laat zien hoe de code tussen '.header' en 'endh' er doorgaans uitziet. De code die daar staat is overigens bindend, andere combinaties dan die opeenvolgingen van '.word' en '.byte' staat de Geoassembler niet toe, de programmeur rest alleen het invullen van de waarden.

```
.header
.word    $00     ;last block -- always 0
.byte    3       ;icon width -- always 3
.byte    21 ;    icon height -- always 21
;icon information follows

         [Seq]

.byte    C64type     ;usually $83
.byte    GEOtype     ;GEOS file type
.byte    GEOstruct   ;GEOS structure type
.word    FileStart   ;load address
.word    FileEnd     ;end of app. address
.word    InitProg    ;init. address
.byte    filename    ;must be 20 bytes
[.byte   ...]        ;optional fields
.endh
```

Illustratie 2

In illustratie 2 komt een aantal zaken voor die de programmeur zelf moet invullen. Het 'C64type' is standaard $83. In het veld 'GEOtype' dient bijvoorbeeld 'DESK_ACC' of 'APPLICATION' te worden ingevuld. Het 'GEOstruct' veld bevat 'SEQ' voor een sequentieel (normaal) programma en 'VLIR' voor een programma dat GEOS-overlays gebruikt. De labels 'FileStart', 'FileEnd' en 'InitProg', bevatten respectievelijk het begin-, eind- en startadres van het programma. Als er voor deze velden nullen ingevuld worden gebruikt GEOS de standaardwaarden, dit zijn respectievelijk $0400, $03ff en $0400. Let wel op: er hoeven hier niet perse absolute adressen te staan, de programmeur kan ook een label uit het programma invullen. Voorwaarde is natuurlijk wel dat het een label betreft dat buiten de programma-file zichtbaar is (zie vorige aflevering). Tenslotte is er het veld 'filename', hierin staat de naam. Buiten de string, in hetzelfde '.byte'-commando, moet de string met nullen tot 20 bytes worden aangevuld. Anders gaat de GeoAssembler gillen. Wil het gebruik van de genoemde namen 'DESK_ACC', 'SEQ' etc. goed gaan, dan moet wel de GEOS-constantenfile geïmporteerd worden (.include geosSym).

Icons

Het icon is een plaatje van 24 bits (= 3 blokjes) breed bij 21 bits hoog. Vooral met de nieuwere versies (1.2 en hoger) van GeoPaint is het handig om een icon in GeoPaint te tekenen, en in een scrap-file weg te schrijven. Deze kan dan op de bekende manier (de paste-optie) in GeoWrite ingelezen worden. Het uitknippen van

een icon is natuurlijk het lastigste karwei. In de versies 1.2 en verder, die met kleur werken, is het handig om met de kleurenstand aan te werken. Scrap-files die dan gesaved worden, rondt GeoPaint altijd op een 8-voud af. De breedte van 24 bits gaat er dan niet zo vaak naast. De lengte is helemaal geen probleem, want als een icon te lang is, dan wordt de rest gewoon weggelaten. U kunt het icon dus rustig een stuk te lang afknippen. Denk er wel om, om de linkerbovenhoek van het icon in de linkerbovenhoek van een kleur-blokje te plaatsen (bijvoorbeeld in de linkerbovenhoek van het scherm). Anders zorgt het zojuist genoemde afronden er voor, dat het altijd misgaat. Een aardig weetje is overigens dat de GeoAssembler voor het werken met plaatjes in de assembly listing 2 speciale variabelen kent: picH en picW. Zij bevatten respectievelijk de hoogte en de breedte in pixels van het laatst voorgekomen plaatje.

Geolinker

De GeoAssembler produceert zogenaamde '.rel' files, deze bestaan uit 3 delen:

° 'relocatable' (verplaatsbare) 6502 machinecode;

° onopgeloste expressies uit het programma;

° alle globale labels en constanten uit het programma.

Als de GeoAssembler een ongedefineerd label of een ongedefineerde constante vindt, dan wordt deze opgevat als een extern label of een externe constante. De GeoLinker linkt verscheidene '.rel' files aan elkaar tot een uitvoerbaar programma. Doorgaans zijn er tenminste 2 '.rel' files: 1 met het programma, en 1 met de header. Voordat er gelinkt kan worden vraagt de GeoLinker de programmeur om een 'command file'.

Command file

In deze file staat geen machinecode, maar informatie over welke files op disk aan elkaar gekoppeld moeten worden en hoe. Bij het linken worden allereerst alle globale labels en constanten uit alle '.rel' files in een cross-reference tabel gezet. Vervolgens gaat de linker voor de tweede keer door alle '.rel' files heen. Nu, met alle globale labels en constanten in een tabel, moeten alle eerst niet oplosbare expressies op te lossen zijn. Zo niet, dan wordt er een foutmelding gegeven. Als GEOS-programmeur staan u dan ook altijd 3 fout-verbeteringsronden te wachten:

° 1) Syntactische fouten die de GeoAssembler ziet;

° 2) Syntactische fouten in labels (bijv. een lettertje fout tikken), die door de GeoLinker gezien worden;

° 3) Semantische (betekenis-) fouten, die u pas opmerkt als u het programma probeert te laten draaien.

De tweede ronde is diegene waar het nu over gaat. Als alle expressies opgelost zijn, wordt de geheugenruimte voor het programma-gedeelte (.psect) en de geheugenruimte voor het data- gedeelte (.ramsect) vastgesteld, en kan het hele programma, header incluis, naar disk geschreven worden.

Directives

In illustratie 3 staat een overzicht van de opeenvolging van de verschillende GeoLinker 'directives' in de command-file. Deze directives zijn: '.output', '.header', '.vlir', '.seq', '.cbm', '.psect' en '.ramsect'. Ik zal ze één voor één even met u doornemen.

Achter '.output' staat de naam van de uiteindelijke, uitvoerbare file, die de GeoLinker dient aan te maken. Op de volgende regel komt het '.header' commando, met daarachter de naam van de headerfile. Deze naam eindigt op '.rel', want zoals we gezien hebben wordt de header ook in de GeoAssembler gemaakt. Wordt het '.header' commando in zijn geheel weggelaten, dan maakt de GeoLinker zelf een default header. De laad- en startadressen wijzen dan gewoon naar de eerste byte van de programmacode (.psect code)! Is het '.output' commando weggelaten, dan wordt de naam 'test' gebruikt.

Het '.vlir' commando is veruit het ingewikkeldste. Allereerst zal ik iets vertellen over VLIR-files. Als programmeur geeft GEOS u de mogelijkheid om een file te splitsen in verschillende modulen. In de VLIR index-tabel op disk staan de pointers naar de diskblokken waar resp. Het 1e, 2e 3e tot en met het 126e module begint. Is zo'n pointer 0,0 dan bestaat het module niet. Het eerste moduul van een VLIR-programmafile zit standaard altijd in het geheugen. De daaropvolgende modulen hoeven niet perse in het geheugen te staan. Het is aan het programma zelf om dat de beslissen. GEOS geeft de programmeur verscheidene routines ter beschikking om gemakkelijk VLIR-records in te laden en weg te schrijven. In de command file worden nu de verschillende modulen samengesteld uit evt. meerdere .rel files per module, en worden de stan-

<u>Geolinker command-file</u>

```
;EERST:
  .output        naam output-file
  .header        headernaam.rel
;DAN:
;-------------------- vlir ---------------------------------
  .vlir          ;een vlir file
  .psect         $xxxx ; begin resident programmagedeelte
  .ramsect       $xxxx ; begin bijbehorend datagedeelte
   1 of meer .rel filenamen
                 ;het residente programmagedeelte
                 ;= het eerste VLIR record in de file

  .mod 1
  .psect         $xxxx ; geheugen voor programma uit mod. 1
  .ramsect       $xxxx ; geheugen voor data uit mod. 1
   1 of meer .rel filenamen
                 ;het eerste overlay gedeelte
                 ;= het tweede VLIR record in de file
  .mod 2
  .psect         $xxxx ; geheugen voor programma uit mod. 2
  .ramsect       $xxxx ; geheugen voor data uit mod. 2
   1 of meer .rel filenamen
                 ;het tweede overlay gedeelte
                 ;= het derde VLIR record in de file
  etc.
;OF:
;-------------------- seq ---------------------------------
  .seq           ;een gewone sequentiele file
  .psect         $xxxx ; beginadres programma
  .ramsect       $xxxx ; beginadres data
   1 of meer .rel filenamen
;OF:
;-------------------- cbm ---------------------------------
  .cbm:          ;een gewoon CBM machinetaal-
                 ;programma zonder header.
                 ; '.header' is dus niet nodig.
  .psect         $xxxx ; beginadres programma
  .ramsect       $xxxx ; beginadres data
   1 of meer .rel filenamen
```

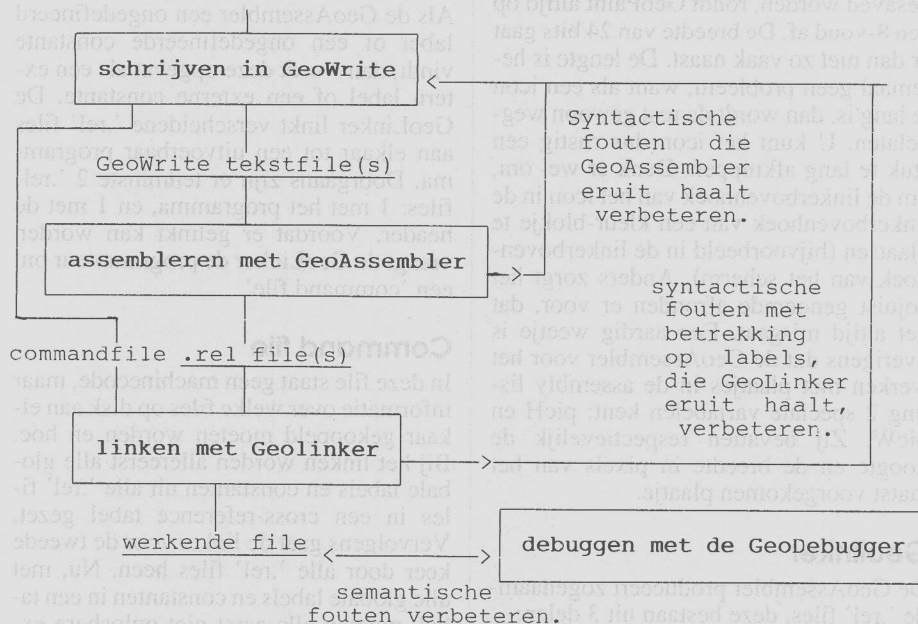
Illustratie 3



daard geheugenplaatsen waar de modulen thuishoren vastgelegd. Na het '.vlir' commando volgt het '.psect $xxxx' commando, de programmacode van de residente module komt dus op $xxxx. Daarna staat het '.ramsect $yyyy' commando. Vervolgens komen een of meer .rel files, deze vormen het residente module. Dan is het de beurt aan eventuele volgende modulen. Deze worden ingeluid met een '.mod x' (x=1,2,3..126) vervolgens weer '.psect' en '.ramsect', en dan de .rel files van de module. Hoewel de modulenummers van 1 tot 126 lopen zijn overigens maar maximaal 20 modulen tegelijk toegestaan. Als overigens ergens het '.psect' commando weggelaten wordt, dan neemt GEOS de standaard waarde $0400 voor het begin van het programma-gedeelte. Is '.ramsect' weggelaten, dan geldt standaard het eindadres van het .pect-gedeelte plus een als het begin van het datagedeelte.

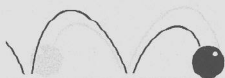
De '.cbm' en '.seq' commando's zijn veel simpeler dan '.vlir'. Zij zorgen er allebei voor dat er een sequentieel (normaal) machinetaalprogramma op disk gezet wordt. Het verschil tussen beide is, dat '.cbm' een file zonder header oplevert, terwijl '.seq' een gewone GEOS-applicatie met header maakt. Alle listings die bij de GEOS-machinetaalcursus zijn verschenen waren gewone '.seq' files. Aan het maken van een VLIR-applicatie zijn we in deze cursus nog niet toe gekomen, maar er zijn nog meer afleveringen op komst!

Peter Boncz



Illustratie 4





Rags to Riches

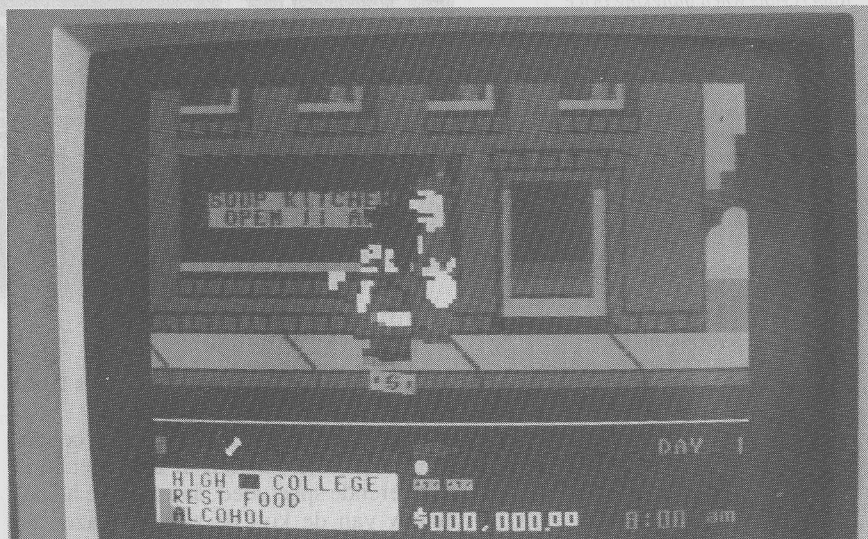
Dit spel benaderd de werkelijk erg goed, de weg naar de top gaat niet over rozen. Voor dat men ook maar enigzins in de buurt komt van een klein kapitaaltje zal men heel wat werk moeten verzetten. En een goede schoolopleiding is daar bij nooit weg. Bij deze uitspraak wordt er niet alleen aan de lagere school gedacht.

We starten het spel in het zuidelijkste deel van de stad. Omdat het spel Engelstalig is spreken we hier bij over South city. Dit deel van de stad is het meest eenvoudige, goedkope deel van de stad. Alle inwoners leven hier ver onder het bestaansminimum. Er is een gaarkeuken waar men van 11.00 uur 's morgens tot 19.00 uur 's avonds gratis een kom soep mag halen. Ook is hier een supermarkt. Het bijzondere hiervan is dat deze 24 uur per dag open is. Dat is wel handig je kan op elk moment van de dag boodschappen doen, maar je kunt hier ook de lege flessen kwijt die er regelmatig in de goot gevonden worden. Behalve deze flessen liggen er ook munten in de goot, en als je geluk hebt vindt je zelfs een dollarbiljet. Mocht dit allemaal niet lukken en begint de honger op te komen, dan is er altijd nog het pandjeshuis. Hier zijn de kaartjes voor de ondergrondse te belenen voor wat voedsel. Twee goedkope hotels horen uiteraard ook thuis in dit deel van de stad. Verder ontbreekt ook niet een drankwinkel, een slager en een kleinere supermarkt. Het grote park wordt onveilig gemaakt door een boosaardige hond. Er bevinden zich een aantal flessen in het park. Dit kan flink wat statiegeld opleveren. Maar let op: de hond is alleen te misleiden met een bot.

Het loont zelfs de moeite om eerst voor een dollar een aantal botten te gaan kopen om zo deze flessen te bemachtigen. Elke keer dat je het park instapt met een bot levert dit maar liefst 5 lege flessen op. Let wel op de sluitingstijd, anders is alle moeite voor niets geweest. Als je tien dollar biljetten heeft gespaard wordt dit bewaard. Het geboefte, dat zich overal in de stad bevindt, kan dit dan niet meer afpakken in dit deel van de stad tenminste. Maar hierover later meer.

Neemt men de moeite om naar de kapper te gaan (dit kost 4 dollar), dan kan men ook een baantje nemen bij de slager. Denk wel om de treinkaartjes, deze kosten 2 dollar per stuk. Je moet dan wel op tijd zijn natuurlijk, als je te laat komt is het baantje al weg. Voor 8 uur 's avonds is het verstandiger om een onderdak te zoeken. Dit stadsdeel is dan zeker niet veilig meer. In één van de twee hotels kost dit 2 dollar per nacht. Behalve de boef die je al je dollar biljetten wil afpakken, loopt er ook een agent rond. Hij pakt allerlei verdachte figuren op. Jammer genoeg behoor jij ook nog tot deze categorie. Zodra hij je ziet, zal hij alles in het werk stellen om je op te pakken. Lukt hem dit, dan verdwijn je in een eenzame cel. Over voedsel hoef je je dan geen zorgen te maken: de voedsel voorraad wordt dan wel aangevuld. Het kan dus best wel eens in je voordeel werken om je te laten oppakken. Als je pas bij de kapper bent geweest zal de agent je negeren en gaat hij op zoek naar het echte gespuis.

We verplaatsen ons even naar een ander deel van de stad namelijk het oosten. Dit is het studiecentrum van de stad. hier vind men de high school, die 's morgens om 8 uur begint. Zorg dat je op tijd bent, anders kom je er niet meer in. Het kost wel enige moeite om alles goed in de gaten te houden. Is het verstandig in zuid te slapen en te profiteren van de gratis soepkeuken?

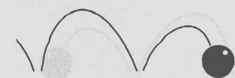


Rags to riches

Het is net te halen, maar een ongeluk zit in een klein hoekje. Kijk zelf maar, wat het beste bevalt. Denk hier ook om je geld, hier heerst ook criminaliteit en zij hebben het voorzien op het grote geld.

In dit deel van de stad kan er een grote, goedkope pizza bestellen voor 1 dollar. Ook zijn er hier campus rooms te huur. Verder is er nog een state college maar hier voor moet 100 dollar neergeteld worden. En dat moet eerst verdient worden.

In het westelijke deel van de stad kan men het geld verdienen. Behalve de al eerder genoemde kapper is er ook nog een bar een gelegenheid "Chez Francois" genaamd, een hot dog tentje en een hamburger restaurant. Kamers zijn hier alleen per week te huur. Zoals gezegd hier kan men het grote geld verdienen. Met een beetje strategie kan men 2 baantjes nemen. Het is dan wel hard werken, maar om verder te komen in deze wereld is geld noodzakelijk. Het noordelijke deel van de stad is letterlijk de top. Hier woont de elite van de zakenwereld. Als je het handig aanpakt kun je ook tot deze groep gaan behoren. Weet je je geld goed te beleggen of zijn anderen je steeds voor. Succes in ieder geval en wie weet wordt jij wel een van die zwervers die het lukt om hier carriere te maken, al is de weg erg lang en zwaar.



Rupert and the ice castle

De bedoeling van dit spel wordt duidelijk gemaakt in het volgende gedicht:

Rupert has to resque Bill whom
Jenny Frost has rozen still
Rupert jumps avoiding jack to
claim his good frind Bingo Back
Icicles fall past moving sledge
as Edward stands on window ledge

Rupert runs when thingking twice
as Jenny aim is not to nice

Rupert sees the bouncing eggs so
resques Bill with bouncing leggs
Rupert slides but must not fall
to resque Bingo from the hall
Edward stands on wall of castle
Rupert climes avoiding hassle
As Rupert ponders when to go
a snowman throws a ball of snow

Together jack and Rupert stick
to put a end off Jenny's trick

Finding objects on his own
Rupert stand on ice kings throne
On toer top stands Badger Bill
so Rupert takes the safest drill

Collecting tings from corridors
Duck jump walk but do not pause
Ice is falling toys are flying
Rupert must not give up trying

Garments Rupert has to find will
hang at top of stairs which wind
Rupert climbs a banquet chair
retrieving Edward from dispair

Snowballs swoop in his direction
can he dodge them in perfection
Rupert has to quicken pace
if algy is to leave this place.

Zoals in vele computerspelen wordt de held gespeeld door de man (of vrouw) achter de joystick. Het is Rupert, de hoofdpersoon in dit verhaal en de redder in nood. Jenny Frost is een kwaaie tante waar je beslist geen sneeuwballen naar moet gaan gooien in de winter, want dit wordt onmiddellijk afgestraft. Zij heeft zo haar eigen manier van spelletjes spelen; alleen vinden anderen dat niet altijd even leuk. Ditmaal zijn de vriendjes van Rupert de slachtoffers. Tijdens de winterse pret heeft Jenny ze meegelokt naar het kasteel van Koning Winter. Eenmaal daar aangekomen zijn ze door Jenny bevroren. Nu is het de taak van Rupert om ze te redden. Lukt dit niet, dan zijn ze voor altijd opgesloten in het winterse paleis.

Zij zijn niet allemaal bij elkaar geplaatst, dat zou te makkelijk worden. Op verschillende plaatsen heeft Jenny ze ingevroren. De één staat bijvoorbeeld op de kasteelmuur, terwijl de ander in het venster staat. Maar ga eerst maar eens op weg. Je ziet dan alles zelf wel. Het grote gevaar in dit uitgestrekte, onoverzichtelijk kasteel van de winterkoning, wordt nog gevaarlijker gemaakt door de verschrikkelijke sneeuwman. Hij is één van Jenny's beste maatjes. Hij gooit regelmatig met sneeuwballen naar je. Wordt je teveel geraakt door deze ballen of door het rond slingerende speelgoed, dan moet je helaas blauw van de kou de aftocht blazen en zijn je vrienden reddeloos verloren.

Al meteen bij het eerste scherm zien we niet alleen Bill boven op de muur staan, maar er zwerven ook twee sleeën rond. Twee glijbanen bewijzen in ieder geval dat we op de goede weg zijn. Koning Winter zou Koning Winter niet zijn als deze sleeën niet behekst waren. Zij glijden dan ook vanzelf, dit alles natuurlijk om Rupert dwars te zitten. Kijk uit voor de al eerder genoemde glijbanen, want daar kun je uiteraard op (uit) glijden. Erg lastig wordt het je in dit eerste level nog niet gemaakt, dat komt later allemaal. Zodra Rupert zijn vriendje aanraakt, ontdooit deze en kan nu door Rupert naar het veilige bos gebracht worden. Als Rupert zijn tweede vriendje Bingo moet gaan redden, maakt hij kennis met Jack, een zeer hupserig dwarsliggertje. Bij aanraking spring je een eind weg en natuurlijk niet in de goede richting. Om het je noch lastiger te maken vallen er af en toe ijspegels. Ook deze kun je beter vermijden, anders is het spel vroegtijdig ten einde. Als ook Edward gered is, maak je kennis met weer een nieuwe tegenstander: de springende eieren. Na een korte kennismaking met Jenny kan Rupert weer opnieuw beginnen met het redden van zijn vriendjes. Alleen staan ze nu op andere plekken en wordt het steeds lastiger om ze te redden. Nieuwe, gevaarlijkere hindernissen maken het hem er niet gemakkelijker op.

Voor het eerst maken we ook kennis met de sneeuwballende gooiende sneeuwman. Als ze allemaal weer gered zijn, kan Rupert nog een keer de stoute schoenen aantrekken. Want behalve dat zijn vriendjes weer gepakt zijn, heeft Jenny nu ook de kleding afgepakt dus moet deze ook door Rupert verzameld worden. Het wordt er nu niet gemakkelijker op want Jenny beseft met een sterke tegenstander te maken te hebben en gooit de wapens op een andere manier in de strijd. Dus blijf allert, kijk goed hoe de ijspegels vallen, de ballen gegooid worden, want er zit een zekere regelmaat in. Zoek bijtijds een veilig plekje en bekijk dit goed. Een goed einde zit er in voor de echte redders en doordouwers onder ons veel succes met deze gouwe ouwe.



Rupert

Lode Runner Resque

Ook in dit spel, Lode Runner Resque moet er weer iemand bevrijd worden. Om dit te kunnen doen, moeten er een (groot) aantal sleutels gevonden worden. Deze zijn niet al te moeilijk te vinden. Her en der liggen ze verspreid en de meeste zie je zo liggen. Een enkeling is verstopt zodat je daar zal dan enig speurwerk voor moet verrichten. In totaal moeten er 2471 sleutels bij elkaar gezocht worden. een aardig klusje dus.

Op het allereerste scherm verschijnt de naam van het programma, een bewaker loopt een vast parcours en is makkelijk te ontwijken. Een kat ligt rustig in een hoekje te slapen. Vang je deze kat, dan krijg je er een extra leven bij. Heb je de eerste vijftig sleutels verzameld, dan gaat er een blokje knipperen daar moet je dan naar toe. Door hierop te gaan staan ga je door naar het volgende scherm. In het tweede level sta je voor een meer met in het midden een eiland. Twee wachters zijn er deze keer, zodra je in hun buurt ver-

schijnt, komen ze op je af. Let er goed op, want een aanraking met is dodelijk.
Bij level vier, Towers, maak je voor het eerst kennis met op en neer gaande hoekjes. Deze zijn noodzakelijk om bij alle sleutels te kunnen. Bij level zeven ben je te klein om sommige hindernissen te nemen. Daarom liggen hier en daar gebakjes op je te wachten, eet je deze op dan wordt je tijdelijk groter, zodat je de eerst onneembare hindernissen wat makkelijker genomen kunnen worden. In het doolhof in level tien zie je de sleutels niet liggen en moet er goed opgelet worden waar je al bent geweest. Het collosseum is erg lastig door de doorzakkende stukjes en de ronde vorm gelukkig geen bewakers die het nog lastiger maken. Een flinke klus staat je te wachten in level 17, 777 sleutels moeten er hier verzameld worden je wordt bijna gek van al dat geknipper. In level 21 is het tijd voor een frisse duik voor je al de 35 sleutels hebt verzameld ben je aardig doorweekt of was je dat al? Taartjes taartjes en nog eens taartjes. Taartjes tot je geen taart meer kan zien. Voor je alle dertien sleutels hebt verzameld, zal je er heel wat achter de kiezen hebben. In level 24 hoeven er geen sleutels verzameld te worden. Alleen de uitgang moet worden gevonden. Level 28 is een grote pyramide: 7 sleutels en 2 bewakers zijn de hoofdmoot. In level 35 staat er weer heel wat anders te wachten: duizelingwekkende hoogtes, watervallen die met razend geweld naar beneden storten. Heb je nog geen genoeg van trappetjes in level 37 zijn er weer genoeg...
Een duister geval level 41: waar zijn de sleutels toch gebleven? Ergens liggen er 15 verborgen. Weer een nat pak in level 44; tot aan je nek kom je in het water te staan, totdat alle 37 sleutels gevonden zijn. En dan eindelijk in level 46 is het einde in zicht. Rechts zie je de te bevrijden persoon in een gevangenis, die streng bewaakt wordt door de rondlopende bewakers. Maar voor het spel echt is afgelopen, moeten er nog wel enige hindernissen genomen worden, De ervaring opgedaan in de vorige levels komt goed van pas, en of er nog speciale trucjes over zijn voor dit laatste level moet je zelf maar ontdekken. Een heel leuk spel door de grote variaties en verrassingen. Een groot voordeel is, dat men zodra men "af" is niet weer helemaal van voren af aan moet beginnen, maar start waar men is gebleven. Men krijgt dan weer drie levens en kan men weer verder gaan. Ook kan je het spel saven, zodat men zelfs de volgende dag weer verder kan gaan, zonder weer helemaal opnieuw te moeten beginnen. Het zal wel enig doorzettingsvermogen kosten om het eind te halen maar het hoeft gelukkig niet allemaal op een dag.

R.G.

GEOS misser

In het vorige nummer van Commodore Info zijn er een aantal illustraties bij de GEOS artikellen weggevallen, deze illustraties worden hier alsnog geplaatst

```
;**************************
;dumpLNK   door Cools Patrick
;**************************
.output         1541-dump
.header         dumpHDR.rel
.seq
.psect          $033c
dump.rel


;**************************
;dumpHDR   door Cools Patrick
;**************************
.if             Pass1
.include        geosSym
.endif
.header
.word           0
.byte           3
.byte           21
                1541

.byte           $80 | USR
.byte           DESK_ACC
.byte           SEQUENTIAL
.word           DAStart
.word           DAEnd
.word           DAStart
.byte           "ScreenDump  V1.0",0,0,0,$00
.byte           "Cools Patrick       ",0
.block          160-117
.byte           "Breng Uw Hardcopy op een "
.byte           "veilige disk anders is ze gewist",0
.endh


;**************************
;dump   door Cools Patrick
;**************************
.if             Pass1
.include        geosSym
.include        geosMac
.endif
.psect
DAStart:
LoadW           r0,filenaam
jsr             DeleteFile
jsr             i_MoveData
.word           data1
.word           fileHeader+68
.word           7
MoveW           SCREEN_BASE-2,data3
LoadW           SCREEN_BASE-2,$2000
LoadW           r9,fileHeader
LoadB           r10L,1
LoadW           fileHeader,filenaam
jsr             SaveFile
MoveW           data3,SCREEN_BASE-2
jmp             RstrAppl
filenaam:
.byte           "DUMP",0
data1:
.byte           $80+ASSEMBLY
.byte           NOT_GEOS
.byte           SEQUENTIAL
.word           SCREEN_BASE-2
.word           SCREEN_BASE+SC_SIZE
data3:
.word           0
DAEnd:
```

GEOS Info illustratie 1

Operators

The following table shows all the valid operators and their precedence:

OPERATOR	PRECEDENCE	
( )	1	grouping parentheses (sub-expression)
-	2	unary negation
!	2	logical not
~	2	bitwise one's complement
[or <	2	low-byte
] or >	2	high-byte
**	3	exponentiation
*	4	multiplication
/	4	division
//	4	modulus
+	5	addition
-	5	subtraction
>>	6	logical shift right
<<	6	logical shift left
>	7	logical greater than
>=	7	logical greater than or equal to
<	7	logical less than
<=	7	logical less than or equal to
= = or =	8	logical equal
!=	8	logical not equal
&	9	bitwise and
^	10	bitwise exclusive-or (xor)
|	11	bitwise inclusive-or (ora)
&&	12	logical and
^^	13	logical exclusive-or
||	14	logical inclusive-or

GEOS Machinetaal illustratie 1

Assembly Control

.include	Include another source code file into the assembly.
.zsect	Begin zero-page section.
.ramsect	Begin unitialized data section.
.psect	Begin program section (default).
.echo	Echo text to the error file.
.end	End assembly (optional at end of source file).

Symbols

=	define equate; do not send symbol to debugger.
= =	define equate; send symbol to debugger.
.eqin	Begin sending equates to the linker (default).
.noeqin	Stop sending equates to the linker.
.glbl	Begin sending global labels to the linker (default).
.noglbl	Stop sending global labels to the linker.

Data

.byte	Include byte-sized data and strings into psect section.
.word	Include word-sized data into psect section.
.block	Reserve space.

Conditional Assembly

.if	Start conditional; assemble if expression is true.
.else	Assemble if expression was false.
.elif	Start new conditional if expression was false.
.endif	End conditional.

GEOS Machinetaal illustratie 2

GEOS INFO

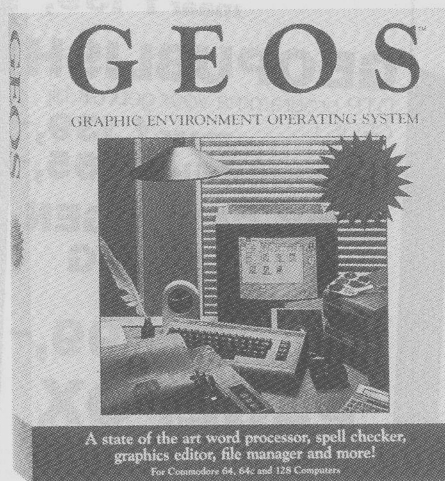
Om mysterieuze redenen heeft een brief er een aantal maanden over mogen doen om mijn bureau te bereiken. De brief is van Joost en Frans H. die toen (september '89) reageerde op deel 6 van de machinetaalcursus, ook geschreven door ondergetekende. Zij vertellen (corrector: vertelden) dat zij de beschikking hebben over GEOS-printerdrivers voor printers die aan de user-poort zitten. Deel 6 van de machinetaalcursus ging over het zelf maken van printerdrivers, het was vooral gericht op het schrijven van een driver voor de gewone seriële poort. In het artikel werden bezitters van drivers voor de user-poort verzocht te reageren. Vandaar dus. De volgende printerdrivers komen in hun lijstje voor:

BlueChip, C.Itoh 8510, C.Itoh 8510A, Gemini 10x. IBM5152+, Epson FX-80, Epson JX-80, Epson MX-80, Oki ML-92/93, PR 2300, Star NB- 15, Star NX-10 en Star SG-10.

Deze printers werken met de zogenoemde GeoCable. Als er met een Geocable gewerkt wordt, komt er in de verbinding tussen de C-64 en de printer geen enkel interface aan te pas, het is een doodgewone kabel tussen de pinnetjes van de user-poort en een aantal van de pinnetjes van de centronics-stekker. Voordeel van deze aanpak is dat dit natuurlijk sneller werkt dan de gewone seriële poort (de data-overdracht gaat nu immers parallel), en dat een GeoCable zelf gemaakt of gekocht veel goedkoper is dan een interface voor centronics-printers aan de seriële poort. Deze dingen kosten al gauw *f* 200,-. Het grote nadeel van de GeoCable is natuurlijk dat er zo weinig drivers voor zijn, maar daar is bij deze dan enige verandering in gekomen.

Opsturen

Joost en Frans zeggen dat mensen die nog op zoek zijn naar de 'ware' (printerdriver), een lege diskette kunnen sturen (met



voldoende retour-postzegels) dan zetten zij hem erop. Het adres is: Kroostweg 59, 3704 EB Zeist. De reden dat ik de achternaam van Joost en Frans niet voluit noemde is niet dat ze gevaarlijke criminelen zijn, maar omdat anders alles en iedereen (je weet maar nooit wie deze rubriek allemaal leest) kan gaan opbellen, en dat zal vast hun bedoeling niet geweest zijn.

Centronics

De drivers van Joost en Frans zijn hoogst waarschijnlijk de oplossing van het probleem van dhr. Middelkoop, die ons schrijft over zijn problemen bij het werkend krijgen van een AVT 80alfa-printer. Dit is een standaard centronics-printer, die aangesloten dient te worden op de user-poort.

Star SG-10C

De tweede brief die zolang onderweg is geweest komt van dhr. Nooyer. Dhr. Nooyer heeft een probleem met zijn STAR SG-10C printer, dit probleem is : hij werkt niet. Het is natuurlijk ondoenlijk voor een eenvoudig GEOS-redacteur om de werking van alle printers voor de C-64 uit zijn hoofd te kennen, en ik weet dan ook niet precies hoe de SG-10C werkt. Wat ik aanraad is om allereerst alle beschikbare drivers te proberen. Schroom ook niet om de gewone 'Commodore' driver uit te proberen, de SG-10C heet toch volledig compatible te zijn met een standaard CBM-printer. Als dit niet helpt, is het waarschijnlijk het beste om zelf, of met behulp van een plaatselijk machinetaal-orakel een eigen driver te schrijven, of er een aan te passen (bijvoorbeeld de Star NX10-C driver, deze zit standaard bij GEOS 1.2 en verder). Het al genoemde deel 6 uit de machinetaalcursus en uw printerhandleiding kunnen u hierin een heel eind op weg helpen.

RAM 1764

De laatste vragensteller van deze maand is tenslotte dhr. Cranendonk. Ik zal me bij hem beperken tot de vragen die met GEOS te maken hebben. Allereerst wil hij weten of GEOS 2.0 werkt met 2x1541 en een RAM-uitbreiding. Het antwoord hierop is: Ja! Het tegenovergestelde antwoord moet ik geven op de vraag of GEOS 2.0 ook werkt op 2x1541 en 1581 (de 3.5" Commodore-diskdrive) en een RAM-uitbreiding bij elkaar. Eenmaal 1541, 1581 en een RAM gaat waarschijnlijk wel, alhoewel ik dit niet zeker weet, want ik heb een 1581 en een RAM nog nooit samen gezien. Het zijn immers beide schaarse artikelen.

Rectificaties

Tenslotte nog wat verdere opmerkingen over het vorige nummer. Het foutenduiveltje heeft weer genadeloos toegeslagen, en mijn machinetaalartikel en de GEOS-Info-rubriek zijn nogal in de soep gelopen. Hierbij geplaatst zijn (hopelijk nu dan wel) de beloofde 1541-dump uit de vorige GEOS-Info-rubriek, en de drie illustraties uit het vorige GEOS-machinetaal artikel. Maanden oude brieven en ook nog dit: we maken niet zo'n beste beurt bij onze lezers. Troost u met de welgemeende verontschuldigingen van de redactie en ondergetekende.

Peter Boncz

Extra Extra Extra Extra Extra

In het hart van dit nummer is een aflevering te vinden van 'geoWorld', het blad voor leden van de Stichting GEOS Gebruikers uit Almere.
De redactie was van mening dat deze insert een welkome aanvulling was op ons blad.
Het gaat hier overigens om een proef.

geoWorld



Jaargang 3, nummer 2

COLOFON

De GEOS NieuwsBrief verschijnt elke twee maanden en is in een oplage van 600 stuks. Een abonnement kost: fl. 35,-- per kalenderjaar.
geoNieuws is een uitgave van de St. GEOS Gebruikers Almere.

Redaktieadres:

Postbus 52,
1300 AB Almere

Tel. spreekuur :

Iedere dinsdagavond van 19:00 uur tot 22:00 uur.

Tel:03240-10041/ 18868

Eindredaktie:
Pim Broekhuijzen

Redaktie:

Coby Smit
Paula van Eijk
Sjef van Egmond

VOORWOORD.

De tweede krant in 1990 is een feit. Het blijkt steeds weer een heel karwei te zijn ondanks de hulp van geoPublish en geoWrite.

Allereerst vindt u een artikel over Public Domain Software. Dit is de tekst zoals hij ook naar de Commodore Info redaktie gaat ter plaatsing. Zo heb ik nog 3 artikelen naar de Commodore inFo gestuurd.
Zoals u wel heeft gelezen is Bert Venema tijdelijk niet aanwezig. Peter Bonz neemt de rubriek tijdelijk waar.
Peter is eigelijk te technisch om de rubriek te kunnen waarnemen, dus vandaar dat ik wat lichte kost heb gestuurd.

Het laatste nieuws is dat Sjef van Egmond voor onze Nieuwsbrief is gaan schrijven. Het artikel BigClipper en Het getal is van zijn hand. Ziet er heel leuk uit. Vindt u ook niet.

Dan is vervolgens de handleiding **geoCalc** klaar en leverbaar.
Het is een 85 pagina tellend boekwerk geworden.
Hoofdstuk 3 is als een cursus geschreven en helpt u inderdaad een heel eind op weg. Dan zijn er ook nieuwe diskettes uit, van ieder een. De PD diskette nr. 7 is geheel op geoPublish gericht met nieuwe lay-outs, grote Fonts en leuke tekeningen. Op de Clip-Art disk nr 9 staan nieuwe borders(lijsten) in geoPaint formaat.
De Fontdisk nr. 9 staat weer vol met geconverteerde PrintFox lettertypes.
Dus er zal best wel wat van uw gading bij zitten.

Dan wil ik u nog even attent maken op de prijsvraag die wij uitschrijven voor het beste ontwerp van een nieuwe voorkant van de krant.

Ook de rubriek **Nog even dit....** heeft weer interessante onderwerpen.

Coby heeft het programma Egyptian Siege uitgeprobeerd en heeft daar een leuk artikel over gemaakt. Dan rest mij alleen te melden dat u de agenda goed moet doorlezen. U krijgt namelijk geen aparte uitnodigingen meer

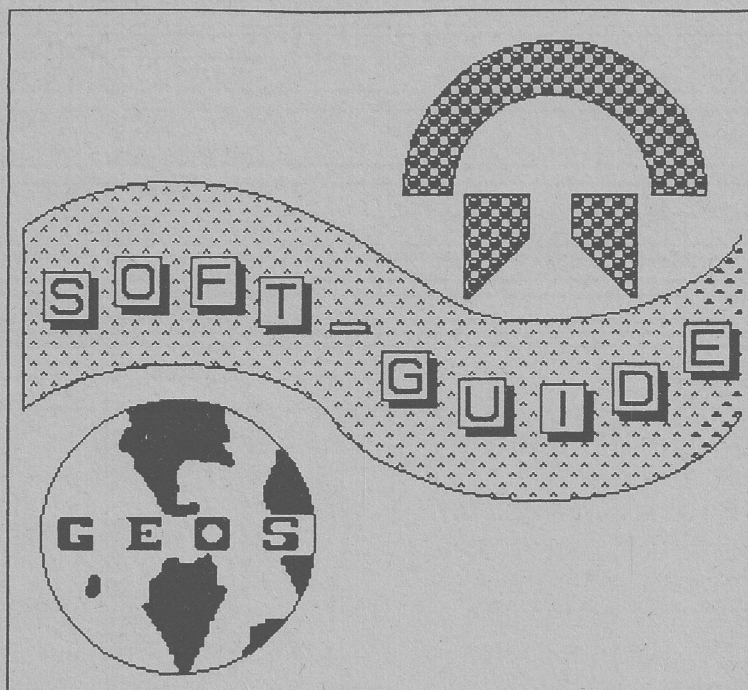
Al met al denk ik toch dat dit een hele geslaagde uitgave van onze krant is.

Pim Broekhuijzen

Inhoud

BigClipper is een programma, dat zodanig ontworpen is, dat hiermee vrijwel elke afmeting aan een PhotoScrap gegeven kan worden. Het doel van deze applicatie is, de beperkingen op te heffen, die door GeoPaint opgelegd worden.

Het Paintscherm levert een scrap op van 144 X 264 pixels. Om nu toch een grotere graphic in een album op te kunnen slaan, moet dit in een aantal stukken weggeschreven worden. Geopaint heeft niet de mogelijkheid, een scrap van b.v. een compleet A4 formaat te **pasten**. Het heeft zodoende ook weinig zin,



om een **BigClipper** scrap in een geoPaint file toe te passen. Het wordt echter heel interressant, als we onze grote graphic (groter dus dan een GeoPaint tekenscherm) naar een geo-Publish- of geoWrite dokument over kunnen schrijven. En dan niet in stukjes, maar de hele afbeelding in een keer!
Met **BigClipper** maak je een scrap, gebaseerd op de pixel-coordinaten van de linkerbovenkant en rechteronderkant van een afbeelding.
Een briefhoofd, aangemaakt in geoPaint kan op deze manier in een keer over de totale breedte in een geoWrite file geimporteerd worden.
Een ander kenmerk van **Big Clipper** is dat de coordinaten, die ingevoerd worden voor de gebruiker, niet perse deelbaar moeten zijn door 8 (vanaf de hoeken).
Een en ander komt uiteraard ten goede aan de flexabiliteit.



MENUBALK

We beginnen uiteraard door op het **BigClipper** icon te dubbelklikken vanuit de DeskTop.
Er verschijnt een menubalk met de volgende toegangsmogelijkheden:
GEOS, DONE, CLIP en HELP.
De GEOS optie werkt zoals we al gewend waren. Door een klik op de tekst kun je je toegang verschaffen tot eventuele Desk Accessoires, die opdezelfde diskette staan.

Kiezen we voor **DONE**, dan stopt het programma en wordt teruggekeerd naar de deskTop.

Bij wijze van beveiliging beschikt **DONE** over slechts 1 submenu, n.l. **QUIT**.
Wilt u dus echt naar de deskTop, klikt u hierop.
Toch niet? Geen nood, gewoon de pointer uit het **QUIT** menu weghalen en het menu sluit zich weer.
Met **CLIP** wordt een clip aangemaakt van een geoPaint afbeelding. Het **HELP** item toont een aantal hulpwindows, met een korte verklaring van de werking van het programma.

CLIP

Nadat er op **CLIP** geklikt is, verschijnt er een window met de namen van de GeoPaint bestanden, die op de schijf staan, net zoals bij het openen van een bestaande file in b.v. geoWrite of geoPaint.

Selecteer nu een Paintfile door erop te klikken (inverteren), daarna klik je op **OPEN**. Met **CANCEL** keer je terug naar het hoofdmenu (de menubalk). Door op **DISK** te klikken, kan een andere GeoPaint diskette uitgelezen worden, zonder naar de desktop terug te hoeven keren.
Ben je in het gelukkige bezit van een tweede diskdrive, klik dan zo mogelijk op **DRIVE**, waarna ook deze diskette in de respectievelijke drive afgetast kan worden.

TOP/BOTTOM

Eenmaal gekozen voor een van de geoPaint datafiles, wordt de gebruiker gevraagd, de boven- en ondercoordinaten in te toetsen.
De KERNAL vraagt om vermenigvuldigingen van 8. Het totale aantal vertikale pixels loopt bij GEOS van 0 tot 720 . Een clip van een afbeelding, die de onderste helft van het Paint-A4 beslaat, wordt als volgt ingetoetst: **360,720.**
Maar ook als we b.v. 358,713 in zouden voeren, dan worden hier automatisch een aantal pixels aan toegevoegd.
Dit aantal pixels wordt eerst netjes op het scherm getoond, zodat de gebruiker kan kiezen of hij het er mee eens is (d.m.v. **OK** of **CANCEL**).

LEFT/RIGHT

Nadat de boven- en onderkant coordinaten zijn aangegeven, vraagt het programma om de linker- en rechtercoordinaten. Ook hier is weer een grens gesteld aan het bereik. Het programma accepteert hier alle invoer tussen 0 en 640.
Alle tussengetallen worden net zo behandeld als in de boven/onder coordinaten.



Voorbeeld voor linkerhelft: 0,320.

BUFFER

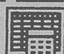
BigClipper beschikt over een 8 Kbyte buffer, waar de clip in opgeslagen wordt.
Toch is het mogelijk, dat de clip teveel geheugen beslaat om hierin een plaats te krijgen. Dit kan gebeuren, als je een groot gebied aangegeven hebt met veel pixels, die 'aan' staan. Ook moet je bedenken bij het binnenhalen van een grotere scrap in geoWrite, of geoPublish, dat deze applicaties zelf ook over een gelimiteerde buffer beschikken. Eerst even uitproberen dus. **BigClipper** houdt geen rekening met eventuele kleurinstellingen.
De informatie wordt opgeslagen in zwart op een grijs3 achtergrond.

SCHAAR

Na de 4 coordinaten ingevoerd te hebben, gaat **BigClipper** aan het werk.
Na verloop van tijd wordt de linkerbovenzijde van de clip op het scherm getoond en afhankelijk van de maat wordt het gehele scherm gevuld.
Tevens bevindt zich nu linksboven een nieuwe menubalk, die het mogelijk maakt in stappen de gehele clip te bezichtigen.
Dezelfde menubalk biedt via de optie **SAVE** de mogelijkheid, de aangemaakte clip als scrap weg te schrijven naar de diskette, waar ook de originele afbeelding staat.
Als laatste optie de menubalk wordt en **RESIZE** geboden. Hierop klikken wil zeggen, nieuwe coordinaten intoetsen, omdat u niet tevreden was met het resultaat.

Vermeldenwaard is verder nog, dat **BigClipper** zowel op de C64 als op de C 128 (64 mode) draait...
BigClipper vindt u op de **PD** disk nr. 7... ▪

GEOCALC FUNCTIE



========================== ==============

ENKELE REKENKUNDIGE FUNCTIES

ABS()

ABS(celnummer). Als het getal in deze cel positief is, zal de functie geen effect hebben.
Een negatief getal evenwel, wordt omgezet in zijn positieve equivalent.
Voorbeeld: =ABS(-41) wordt =ABS(41).

INT() en RND()

INT(celnummer). Converteert een decimaal getal naar het grootst mogelijk INTeger getal.
Voorbeeld: =INT(14.65) wordt 14, =INT(14.2) wordt ook 14.
RND converteert een decimaal getal naar het dichtstbijzijnde intergere getal.
Voorbeeld: =RND(14.65) wordt 15, =RND(14.2) wordt 14.

RAND()

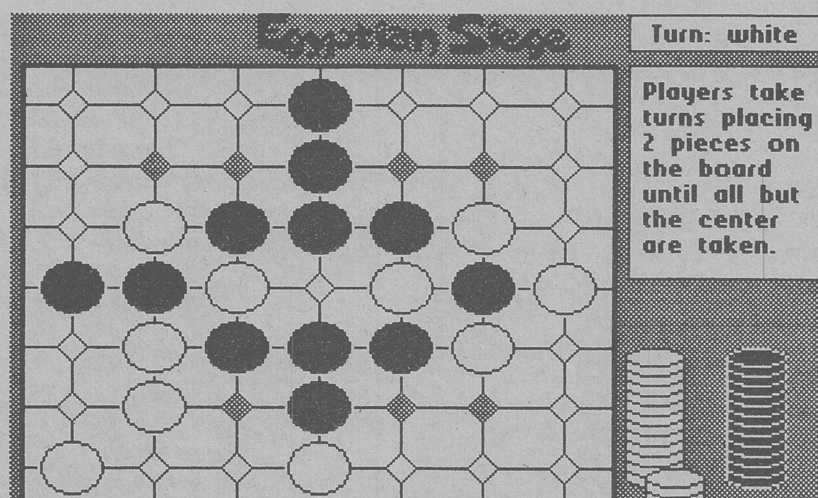
Genereert een getal tussen 0 en 1. Deze functie accepteert geen rekenkundig argument.

SQRT()

Berekent de wortel van een getal.
Voorbeeld: =SQRT(16) geeft als uitkomst 4. Als de uitkomst negatief is, geeft deze functie een *RANGE* error.

Egytian Siege

Egyptian Siege is een spel dat gespeeld kan worden door twee personen of door een persoon tegen de computer. Zodra het spel is gestart komt er een dialoogvenster op het scherm waar u kunt kiezen uit een enkel spel of een dubbel spel. Het veld van een enkel spel is 5x5 cm en van een dubbel spel 7x7 cm groot. Als u gekozen heeft voor een enkel spel dan krijgt u een leeg veld tot u beschikking waarop u 12 stenen kunt plaat- sen naar eigen keuze (in het dubbele spel 24 stenen). Indien u tegen de computer speelt, zal deze als eerste twee stenen plaatsen. Hierna kunt u 2 stenen plaatsen enz. tot het veld gevuld is. Dan gaat het spel beginnen. (wit begint) Het is nu de bedoeling dat u de stenen van uw tegenspeler dusdanig vastzet zodat u ze kunt pakken. Het is mogelijk om de stenen **horizontaal** en



vertikaal over het veld te schuiven. **Diagonaal schuiven is niet mogelijk**. U kunt u stenen verplaatsen door het aanklikken van de steen. U zult dan op de steen die u wilt verplaatsen, pijltjes aantreffen die aangeven welke kant u op kunt gaan. Let wel op dat u niet tusen twee stenen van uw tegenspeler komt te staan, want dan bent u deze stenen kwijt. Mocht u bij het zetten van een steen een steen van uw tegenspeler kunnen pakken, dan staat aan de rechterzijde van het scherm: **End Capture** en een pijltje welke kant u op moet. Het is ook mogelijk dat u meer dan 1 steen kunt pakken. Is dit het geval dan zal er aan de rechterzijde komen te staan: **You may make a multiply capture or choose CANCEL.** U kiest alleen voor **Cancel** als het niet goed uitkomt om de steen van de tegenspeler te pakken. Het is namelijk heel goed mogelijk dat u twee stenen pakt terwijl uw tegenspeler of de computer er vier pakt. In zo'n geval kiest u voor **Cancel**. Het is ook nog mogelijk dat bij het plaatsen van de 12 of 24 stenen, een van beide partijen niet kan zetten. In zo'n geval zult u aan de rechterzijde zien staan: **Stalemate. Black(White) can't move!** In dit geval zal u een nieuw spel moeten starten. **COBY**

Printer Perikelen

Met deze rubriek hoop ik eigelijk zoveel mogelijk gegevens te verzamelen over allerlei printers en hun aansturing. Voorts kan ik in deze rubriek de nieuwe printer drivers vermelden zodat diegene die met een bepaalde printer zittten ook deze informatie tot zijn beschikking krijgt.

MPS 1000
Tot nu toe scheen de MPS 1000 altijd problemen te hebben gegeven met de kwaliteit van de afdruk.
Die tijd is nu voorbij....
Sluit uw MPS 1000 parallel aan op de userpoort van uw C64 of C128 met een **geoCable**.
Tevens dienen alle Dip-switches op 1-1 te staan.
Neem vervolgens het programma Printer Creator en de driver "!!Com1(gc)" en plaats deze samen met de deskTop op een lege diskette of op de REU.
Start Printer Creator en open als printerdriver de "!!Com1 (gc)". Als het kommandomenu op het scherm verschijnt kiest u uit het menu "Printer aanpassen" de volgende opties en voert de waarden in die in onderstaande tabel zijn aangegeven.

Aanslag	3
Apparaatadres	3
Sekundair adres	128
Wagenterugloop	10
Paginadoorloop	12
Paginabreedte	13,27,74,21
Initialisering	27,64
Graphic mode	27,51,1,27,90

Save daarna de nieuwe driver onder een andere naam op disk.

Star LC 10 /NX 1000
Voor de Star LC/NX zijn inmiddels al heel wat drivers gemaakt. Toch kwam ik een opmerkelijke 240 dpi driver tegen. Nu wordt de hoogste resolutie momenteel alleen door een laserprinter gehaald met 300 dpi. Het enige nadeel van deze nieuwe driver is, dat door het vele rekenwerk wat gedaan moet worden, hij traag is. Voor de rest heeft deze driver alleen maar voordelen. Nog een grappig iets van deze printerdriver is, dat hij in twee uitvoeringen is nl. 1 met input d.m.v. de pieper van de printer en een met input via een gekleurd vierkantje op het scherm. De invoer van instruktie geschiedt niet via een dialoog box zoals wij dat bij GEOS gewend zijn maar rechtstreekts. Zogauw u een piepje of gekleurd vierkantje ziet geeft u een opdracht. U moet in totaal 4 opdrachten opgeven: 1. Font of Dpi,
2. Pagina lengte, 3. Karakterbreedte (Pitch) en ten 4e. Proportioneel printen. Het uitvoeren van de opdrachten gaat als volgt: na het eerste piepje of bij het eerste gekleurde vierkantje dat op het scherm verschijnt voert u een cijfer in. Deze cijfers worden hieronder uitgelegd.

1. Font of Dpi

<u>TOETS</u>	<u>ASCII (Draft mode)</u>	<u>GRAPHICS</u>
CR =	Courier(standaard)	80 dpi(standaard)
1 =	Sanserif	240 dpi
2 =	Orator(small caps)	80 dpi doublepass
3 =	Orator(lower case)	80 dpi triplepass
4 =		72 dpi

(CR) staat voor <RETURN>

2. Pagina lengte
CR =11" (standaard)
1 = 1"
2 = 3"
3 = 3,5"

3. Karakterbreedte(pitch)
CR = Pica (standaard)
1 = Elite
2 = Pica condensed
3 = Elite condensed

4. Proportioneel printen
CR = **Niet** Proportioneel
1 = Proportioneel

Tevens zijn er nog enkele kommando's die samen met de C= toets nog een funktie in de Draft mode kunnen uitvoeren:
C= / aan/uit "double-strike" ascii
C= @ aan/uit "onderstrepen"
C= * aan/uit "italics"
C= [aan/uit "dubbele hoogte"

Vraag naar deze nieuwe drivers.

Nog even dit...

geoSpell woordenboek

In de vorige krant vroegen wij naar een Nederlands woordenboek.
Daar is wel enkele reaktie op geweest maar niet met de juiste informatie.
Er zijn toch duidelijke voorwaarden waaraan een woordenboek met voldoen.
In het woordenboek komen alle zelfstandige naamwoorden, voorvoegsels, bijwoorden etc. En van de werkwoorden: de stam, de verleden tijd en het voltooid deelwoord. Meer mag er niet in omdat anders de betrouwbaarheid in het geding komt. GeoSpell kan het verschil tussen "word" en "wordt" niet aan en dit zal dan ook aldoor de gebruiker gedaan moeten worden.

HP Deskjet

Met groot genoegen kunnen wij u mededelen dat er door de Stichting een HP Deskjet printer is aangeschaft. Deze printer levert laserkwaliteit.
Hiermede kan de langverwachte LaserPrintService van start gaan.
Op de bijeenkomsten is de printer aanwezig en kan een ieder daar gebruik van maken om dokumenten te printen.
Omdat de tonercartridges nogal aan de dure kant zijn vragen wij een kleine vergoeding voor papier en toner: 20 cent per A4.
Ik denk dat voor niemand een bezwaar zal zijn.
Dus neem een mooie tekening van u zelf mee en print deze op de Deskjet uit. U zult versteld staan van de kwaliteit.

PRIJSVRAAG

Met de nieuwe ideeen die opkwamen voor de krant werd ons ook duidelijk dat er een nieuwe voorpagina voor de krant gemaakt moest worden.
De pagina met het toetsenbord en de naam geoWorld is geleend van een Amerikaans blad. Nu was dit in principe voor een enkele keer gepland, maar de praktijk bewees anders. Nu leek het ons een leuk idee als we daar een prijs aan verbonden. U ontwerpt een nieuwe voorpagina. De enige voorwaarde is dat het een geoPaint tekening is. De naam staat ook vast: **geoNieuws**
De prijs die wij er aan verbinden is een GEOS applikatie naar keuze. Uw eventuele inzending moet voor 20 mei in ons bezit zijn. **Succes.**

RESET knop

Heeft u ook gemerkt dat nu u een RAM 1764 gebruikt u uw RBoot niet kunt gebruiken omdat u niet kunt resetten. Dus moet u de computer toch helemaal uitzetten met het gevolg dat u toch alles uit de RAM kwijt bent. De oplossing voor dit probleem is een RESET knop. Dit is een knopje dat aan de achterkant van de computer boven de Userpoort wordt gemonteerd. Er wordt een gaatje van 4,5 mm geboord in de kast en daar wordt de knop in vastgezet. Voorts worden er twee draadjes aan de knop gesoldeerd en de andere kant van deze draadjes worden gesoldeerd aan pin 1 en pin 3 van de Userpoort. De kast wordt weer dicht gemaakt en voila. 1 reset knop.
Loopt u nu vast met GEOS. GEEN NOOD. Druk op de resetknop. Doe de Bootdisk in de drive en laad het programma RBoot, prg. De deskTop verschijnt op het scherm en u kunt de inhoud van uw RAM veiligstellen. U heeft na een reset maar twee drives tot uw beschikking.
Succes.

Harddisk voor de C64/C128

Het Amerikaanse bedrijf CMD heeft voor de C64 en C128 een stand-alone Harddisk ontwikkeld.
Deze harddisk is niet veel groter als de reeds bekende 1581 3-1/2" drive.
Er zijn drie versie n.l. een 20 megabyte, een 40- of een 100 Mb. En er is een upgrade naar 200Mb.
Hij bezit tevens vier operation modes: 1541, 1571 en 1581 mode en een GEOS of CPM mode. Dit alles zit ingebakken.
De interface is er zowel in een seriele als een parallel uitvoering en werkt met de meeste speeders. De prijs is best redelijk $ 599.95 voor de 20MB, $799.95 voor de 40 en $1299.95 voor 100 Mb.

April 1990						
S	M	T	W	T	F	S
1	2	3	4	5	6	7 *
8	9	10	11	12	13	14 *
15	16	17	18	19	20	21
22	23	24	25	26	27	28 *
29	30					

Bijeenkomsten in April 1990

7 April in Apeldoorn

gebouw Rigolo's
Eksterweg 71
Apeldoorn

van 11:00 tot 16:00

14 april in Amsterdam

Computer inFo beurs
Deltahal RAI complex
van 10:00 tot 17:00 uur

28 april in Almere

Bijeenkomst Almere
buurthuis Meenten & Grienden
Jaagmeent 180
Almere - Haven
van 11:00 tot 16:00 uur

Bijeenkomsten in MEI 1990

5 mei in Belgie

Bedrijfskantine van
ATLAS COPCO AIRPOWER
Boomsesteenweg 957
Wilrijk - Antwerpen
Belgie
van 11:00 tot 16:00 uur
onder voorbehoud

19 mei in Purmerend

Bijeenkomst Computerclub Waterland
Wijkcentrum "De Trekschuit"
Leeghwaterpark 7
Purmer - Noord Purmerend
van 11:00 uur tot 16:00 uur.

26 mei in Almere vervalt i.v.m Hemelvaartdag verschuift naar 9 juni

May 1990						
S	M	T	W	T	F	S
		1	2	3	4	5 *
6	7	8	9	10	11	12
13	14	15	16	17	18	19 *
20	21	22	23	24	25	26
27	28	29	30	31		

St. GEOS Gebruikers
Postbus 52, 1300 AB Almere
tel: 03240-10041/18868
AMROBank 49.80.29.816 Postgiro AMRO 32750

Dinsdagavond telefonisch Geos spreekuur van 19.00-22.00 uur

Print out C-64 met engels

Engels

Nu de examentijd weer voor vele scholieren in het zicht komt, rijst ook de vraag weer om de computer bij het huiswerk te kunnen gebruiken. Stephan Bargeman uit Meerkerk heeft een programma geschreven dat aan deze wens tegemoet komt. Wanneer u het programma opstart, wordt er gevraagd of er een uitleg moet worden gegeven. Na het opstarten moet u even geduld hebben, alle woorden worden dan ingelezen. De computer kan 664 verschillende woorden overhoren. Moet er tussendoor gestopt worden, om bijvoorbeeld eerst even verder te leren, dan moet het woord STOP worden ingetoetst.

```
1    rem *** maker : stephan bargeman *
     **
2    rem ***                          *
     **
3    rem ***        meerkerk          *
     **
4    rem ***   Commodore Info 1990   *
     **
10   rem *** beginscherm ***
20   clr:b=0:f=0:i=0:t=1800:dime$(t):di
     mn$(t):dimc(t)
30   poke53280,0:poke53281,0:printchr$(
     14):printchr$(8)
40   print"[SHIFT-CLR][CRSR-DOWN]":fori
     =1to7:print"[SPACE][CTRL-3][CTRL-9
     ][38xSPACE]":next
50   fori=1to7:print"[SPACE][CTRL-2][CT
     RL 9][38xSPACE]":next
60   fori=1to7:print"[SPACE][COM-7][CTR
     L 9][38xSPACE]":next
70   print"[CTRL-8][CTRL-9][11xCRSR-UP]
     [4xCRSR-RIGHT]E[5xSPACE]N[5xSPACE]
     G[5xSPACE]E[5xSPACE]L[5xSPACE]S"
80   d$="[10xSPACE]Instrukties[SPACE]J/
     N[SPACE]?[8xSPACE]"
90   print"[6xCRSR-DOWN]"
100  print"[CRSR-UP][2xCRSR-RIGHT][CTRL
      9]";d$:forx=1to79:next
110  d$=d$+left$(d$,1):d$=mid$(d$,2,36)
120  geta$:ifa$<>"j"anda$<>"n"then100
130  ifa$="j"then160
140  goto310
150  rem *** instrukties ***
160  print"[SHIFT-CLR][CTRL-2][CTRL-9][
     15xSPACE]INSTRUKTIES[14xSPACE][CTR
     L 0]"
170  print"[2xCRSR-DOWN]De[SPACE]comput
     er[SPACE]geeft[SPACE]een[SPACE]Ned
     erlands[SPACE]woord."
180  print"[CRSR-DOWN]Als[SPACE]je[SPAC
     E]het[SPACE]goed[SPACE]hebt[SPACE]
     krijg[SPACE]je[SPACE]er[SPACE]een"
190  print"punt[SPACE]bij."
200  print"[CRSR-DOWN]Als[SPACE]je[SPAC
     E]het[SPACE]fout[SPACE]hebt[SPACE]
     krijg[SPACE]je[SPACE]het[SPACE]woo
     rd"
210  print"later[SPACE]nog[SPACE]een[SP
     ACE]keer."
220  print"[CRSR-DOWN]De[SPACE]computer
     [SPACE]overhoort[SPACE]alle[SPACE]
     woorden[SPACE]die"
230  print"in[SPACE]de[SPACE]computer[S
     PACE]zitten."
240  print"[CRSR-DOWN]Wil[SPACE]je[SPAC
     E]eerder[SPACE]stoppen,[SPACE]tik[
     SPACE]dan[SPACE]'stop'"
250  print"in[SPACE]als[SPACE]de[SPACE]
     computer[SPACE]het[SPACE]Engelse[S
     PACE]woord"
260  print"wil[SPACE]weten."
270  print"[3xCRSR-DOWN][10xSPACE][CTRL
      9]Tik[SPACE]'RETURN'[CTRL-0]"
280  geta$:ifa$<>chr$(13)then280
290  ifa$=chr$(13)then310
300  rem *** inleesroutine ***
310  print"[SHIFT-CLR][CTRL-8][CTRL-9][
     11xSPACE]Engels[SPACE]Woordenboek[
     11xSPACE][CTRL-0]"
320  print"[CTRL-2][HOME][3xCRSR-DOWN]B
     EN[SPACE]WOORDEN[SPACE]AAN[SPACE]H
     ET[SPACE]INLEZEN[SPACE]!!"
330  fori=1to664:readn$(i),e$(i):t=i:ne
     xti:goto370
340  print"[3xCRSR-DOWN]Tik[SPACE]'[CTR
     L 9]RETURN[CTRL-0]'"
350  geta$:ifa$<>chr$(13)then350
360  rem *** hoofdprogramma ***
370  b=b+1:print"[SHIFT-CLR]In[SPACE]di
     t[SPACE]woordenboek[SPACE]staan";t
     ;"woorden.":print
380  print"Beurt[SPACE]no.:";b:print"[3
     xCRSR-DOWN]Geef[SPACE]het[SPACE]ju
     iste[SPACE]Engelse[SPACE]woord[SPA
     CE]voor[SPACE]:"
390  i=int(rnd(0)*t)+1:forj=itot:ifc(j)
     =1thennext:i=1:goto390
400  print:print"[9xSPACE][CTRL-9]";e$(
     j):ifc(j)=2thenprint"[CTRL-9][CTRL
      8](herhaling[SPACE]!)[CTRL-2]"
410  x$="":poke19,64:poke207,0:poke204,
     1:print"[9xSPACE]";:inputx$:ifx$="
     "then410
420  ifx$="stop"then460
430  ifx$=n$(j)thenc(j)=1:print:print:p
     rint"goed[SPACE]zo[SPACE]!":goto45
     0
440  f=f+1:c(j)=2:print:print:print"fou
     t[SPACE]![SPACE]het[SPACE]moet[SPA
     CE]zijn[SPACE]:[SPACE]";n$(j)
450  print:print"aantal[SPACE]fouten[SP
     ACE]:[SPACE]";f:ifb<t+fthen340
460  print"[SHIFT-CLR]Je[SPACE]hebt[SPA
     CE]nu[SPACE]alle[SPACE]woorden[SPA
     CE]gehad[SPACE]of[SPACE]je"
470  print"wilde[SPACE]stoppen,[SPACE]s
     peler"
480  print"Je[SPACE]vertaalde";(b-1)-f;
     "goede[SPACE]woorden[SPACE]in";b-1
     :print"beurten."
490  print:print"aantal[SPACE]fouten[SP
     ACE]:";f:print:end
500  rem ==============================
     = 505 rem ========= data woorden =
     =======
510  rem ==============================
     =
520  databody,lichaam,face,gezicht,pret
     ty,knap,cheek,wang,to kiss,kussen,
```

```
    kiss,kus
530 databeautiful,mooi,to see,zien,eye
    brow,wenkbrauw,part,deel,leg,been
540 datafoot,voet,toe,teen,to stand,st
    aan,elbow,elleboog,to break,breken
550 datato fall,vallen,knee,knie,mouth
    ,mond,tongue,tong,tooth,tand,hole,
    gat
560 datawrist,pols,nail,nagel,to cut,s
    nijden,to cut,knippen,scissors,sch
    aar
570 dataankle,enkel,hair,haar,head,hoo
    fd,to shake,schudden,thumb,duim
580 datalittle finger,pink,skin,huid,n
    egro,neger,race,ras,throat,keel
590 datalung,long,air,lucht,to hit,rak
    en,beard,baard,below,onder,chin,ki
    n
600 datacolour,kleur,light,licht,betwe
    en,tussen,forehead,voorhoofd,blow,
    klap
610 datato hold,vasthouden,through,doo
    r,tall,lang,soap,zeep,or,of,blood,
    bloed
620 datato lie,liggen,back,rug,shoulde
    r,schouder,to comb,kammen,comb,kam
630 datayear,jaar,month,maand,january,
    januari,february,februari,march,ma
    art
640 dataapril,april,may,mei,june,juni,
    july,juli,august,augustus
650 dataseptember,september,october,ok
    tober,november,november,december,d
    ecember
660 dataseason,seizoen,to start,beginn
    en,spring,lente,summer,zomer,autum
    n,herfst
670 datawinter,winter,last,laatste,wee
    k,week,day,dag,hour,uur,monday,maa
    ndag
680 datatuesday,dinsdag,wednesday,woen
    sdag,thursday,donderdag,friday,vri
    jdag
690 datasaterday,zaterdag,sunday,zonda
    g,to go,gaan,morning,ochtend,from,
    van
700 dataafternoon,middag,evening,avond
    ,night,nacht,today,vandaag
710 datayesterday,gisteren,tomorrow,mo
    rgen,time,tijd,minute,minuut,nearl
    y,bijna
720 dataoften,vaak,last,vorige,since,s
    inds,to come,komen,at once,onmidde
    lijk
730 datawatch,horloge,ever,ooit,never,
    nooit,every,ieder,once,eens,twice,
    tweemaal
740 datatonight,vanavond,during,tijden
    s,midnight,middernacht,sometimes,s
    oms
750 dataearly,vroeg,again,weer,ill,zie
    k,because,omdat,health,gezondheid
760 datawhen,wanneer,pain,pijn,to last
    ,duren,illnes,ziekte,serious,ernstig
770 datato stay,blijven,had better,kun
    t beter,medicine,medicijn
780 datanurse,verpleegster,sick,ziek,s
    ick,misselijk,people,mensen
790 datahospital,ziekenhuis,wound,wond
    ,to look,kijken,to bleed,bloeden
800 datapale,bleek,to feel,voelen,to b
    eat,slaan,toothhache,kiespijn,deaf
    ,doof
810 datadentist,tandarts,to hurt,pijn
    doen,headhache,hoofdpijn,so,dus
820 datahandkerchief,zakdoek,to die,st
    erven,dead,dood,afraid,bang,life,l
    even
830 dataoperation,operatie,to save,red
    den,weak,zwak,family,gezin,large,g
    root
840 datadaughter,dochter,son,zoon,to t
    ake,nemen,only,slechts,like,als,un
    cle,oom
850 dataaunt,tante,age,leeftijd,small,
    klein,parent,ouder,nephew,neef,nie
    ce,nicht
860 datacousin,neef,cousin,nicht,to ca
    ll,noemen,to marry,trouwen,husband
    ,man
870 datamr,mijnheer,mrs,mevrouw,miss,j
    uffrouw,young,jong,too,te,lady,dame
880 datagentleman,heer,friend,vriend,g
    irlfriend,vriendin,couple,paar,cha
    p,kerel
890 databirthday,verjaardag,party,fees
    t,nice,leuk,present,cadeau
900 datagift,geschenk,many,veel,guest,
    gast,to welcome,verwelkomen,welcom
    e,welko
910 datato buy,kopen,to thank,danken,t
    o invite,uitnodigen,card,kaart
920 datato send,sturen,always,altijd,t
    rouses,broek,to wear,dragen
930 datajeans,spijkerbroek,skirt,rok,t
    o make,maken,cotton,katoen,pocket,
    zak
940 datashirt,overhemd,clothes,kleren,
    button,knoop,coat,jas,quality,kwal
    iteit
950 datapyjamas,pyjama,jacket,colbert,
    leather,leer,real,echt,waterproof,
    waterdicht
960 datasize,maat,stocking,kous,wool,w
    ol,woollen,wollen,sweater,trui
970 datashort,kort,under,onder,underwe
    ar,ondergoed,suit,pak
980 datamaterial,materiaal,hat,hoed,sh
    oe,schoen,pair,paar,dirt,vuil,dirt
    y,vies
990 dataclean,schoon,to clean,schoonma
    ken,brush,borstel,dress,jurk
1000 dataglove,handschoen,to keep,houde
    n,food,eten,soup,soep,chicken,kip
1010 datato like,houden van,butter,bote
    r,bread,brood,to bake,bakken,loaf,
    brood
1020 dataroll,kadetje,slice,snee,cheese
    ,kaas,sandwich,dubbele boterham
1030 datato eat,eten,thick,dik,thin,dun
    ,to cook,koken,cook,kok,exelent,ui
    tstekend
1040 datafish,vis,to fry,braden,to smel
    l,ruiken,meat,vlees,to sell,verkopen
1050 databutcher,slager,salt,zout,peppe
    r,peper,grocer,kruidenier,meal,maa
```

- Print out - Print out - Print out - Print out - Print out -

```
     ltijd
1060 databreakfast,ontbijt,lunch,middag
     eten,ready,klaar,to want,willen
1070 datasupper,avondeten,quick,snel,en
     ough,genoeg,bone,bot,egg,ei,to boi
     l,koken
1080 databacon,spek,to mix,mengen,cake,
     taart,piece,stuk,delicious,heerlijk
1090 datadinning-room,eetkamer,hunger,h
     onger,chocolate,chocolade,knife,mes
1100 datasharp,scherp,point,punt,fork,v
     ork,spoon,lepel,plate,bord
1110 datasaucer,schoteltje,cupboard,kas
     t,fridge,koelkast,table,tafel,to s
     it,zitten
1120 dataround,rond,waiter,ober,to serv
     e,bedienen,service,bediening
1130 datamenu,menukaart,to bring,brenge
     n,to order,bestellen,wine,wijn
1140 datadecide,besluiten,to taste,proe
     ven,to taste,smaken,to pay,betalen
1150 datamanner,manier,to stare,staren,
     to smoke,roken,cigar,sigaar,smoke,
     rook
1160 dataplease,alstublieft,vegetable,g
     roente,cauliflower,bloemkool
1170 datagreengrocer,groenteboer,onion,
     ui,banana,banaan,yellow,geel,ripe,
     rijp
1180 datapea,erwt,fresh,vers,tomato,tom
     aat,strawberry,aardbei,patato,aard
     appel
1190 dataorange,sinaasappel,juice,sap,t
     o grow,groeien,pear,peer,bean,boon
1200 datacarrot,wortel,peach,perzik,sof
     t,zacht,coffee,koffie,nice,lekker
1210 datato drink,drinken,drink,drank,g
     lass,glas,to fill,vullen,tea,thee
1220 datasugar,suiker,sweet,zoet,lemona
     de,priklemonade,cream,room,thirst,
     dorst
1230 datapub,cafe,to let,laten,beer,bie
     r,bottle,fles,cork,kurk,another,no
     g een
1240 datadrunk,dronken,to live,wonen,to
      tell,vertellen,same,zelfde,room,k
     amer
1250 dataroom,ruimte,neighbour,buur,kit
     chen,keuken,chimney,schoorsteen
1260 datafloor,vloer,to drop,laten vall
     en,floor,verdieping,roof,dak,flat,
     plat
1270 dataown,eigen,to build,bouwen,buil
     ding,gebouw,lamp,lamp
1280 datato knock,kloppen,to enter,binn
     engaan,entrance,ingang,to push,duwen
1290 datato pull,trekken,to ring,bellen
     ,bell,bel,button,knop,wall,muur
1300 datawindow,raam,to shut,sluiten,ba
     th,bad,bathroom,badkamer,shower,do
     uche
1310 datatoilet,toilet,hall,hal,chair,s
     toel,confortable,gemakkelijk
1320 datafurniture,meubelen,wood,hout,w
     ooden,houten
1330 dataceiling,plafond,to touch,aanra
     ken,bookcase,boekenkast,shelf,plank
1340 datacarpet,tapijt,garden,tuin,behi
     nd,achter,fire,vuur,match,lucifer
1350 datato burn,branden,coal,kolen,hea
     t,hitte,to paint,schilderen,to han
     g,hangen
1360 databedroom,slaapkamer,stairs,trap
     ,curtain,gordijn,pink,roze
1370 datato move,verplaatsen,blanket,de
     ken,tidy,netjes,sheet,laken,to sle
     ep,slapen
1380 datasleepy,slaperig,as,daar,awake,
     wakker,dream,droom,to dream,dromen
1390 datapillow,kussen,town,stad,city,s
     tad,square,plein,capital,hoofdstad
1400 datachurch,kerk,tower,toren,high,h
     oog,busy,druk,to lead,leiden
1410 datato walk,lopen,walk,wandeling,p
     avement,trottoir,save,veillig
1420 datanear,bij,to cross,oversteken,c
     inema,bioscoop,village,dorp,traffi
     c,verkeer
1430 dataroad,weg,way,weg,to drive,rijd
     en,fast,snel,driver,chauffeur
1440 datacareful,voorzichtig,left,links
     ,danger,gevaar,dangerous,gevaarlijk
1450 dataslowly,langzaam,bike,fiets,to
     cycle,fietsen,helmet,helm,to ride,
     rijden
1460 datamotorway,autosnelweg,accident,
     ongeluk,to happen,gebeuren,result,
     gevolg
1470 datato kill,doden,speed,snelheid,d
     irection,richting
1480 datawheel,wiel,tyre,band,to turn,d
     raaien,to repair,repareren,brake,rem
1490 datamechanic,monteur,empty,leeg,pe
     trol,benzine,still,nog,oil,olie
1500 dataengine,motor,to check,controle
     ren,to brake,remmen,suddenly,plots
     eling
1510 datafront,voorkant,seat,stoel,belt
     ,gordel,famous,beroemd,noise,lawaai
1520 datatransport,vervoer,train,trein,
     reilway,spoorweg,to leave,vertrekken
1530 dataplatform,perron,passenger,pass
     egier,to wait,wachten,next,volgende
1540 datato return,terugkeren,ticket,ka
     artje,price,prijs,such,zulk
1550 datagoods,goederen,journey,reis,un
     derground,ondergrondse,to catch,halen
1560 dataconductor,conducteur,to show,l
     aten zien,airport,luchthaven
1570 dataairline,luchtvaartmaatschappij
     ,to fly,vliegen,pilot,piloot
1580 datato arrive,aankomen,arrival,aan
     komst,flight,vlucht
1590 datato travel,reizen,holidays,vaka
     ntie,country,land,foreign,buitenlands
1600 dataforeigner,buitenlander,to stay
     ,logeren,tired,moe,to reach,bereiken
1610 datato book,bespreken,suitcase,kof
     fer,heavy,zwaar,to lift,opptillen
1620 datastrength,kracht,to carry,drage
     n,bag,tas,information,inlichtingen
1630 dataluggage,bagage,trip,reisje,to
     enjoy,genieten van,castle,kasteel
1640 dataguide,gids,campsite,kampeerter
     rein,nationality,nationaliteit
1650 datato allow,toestaan,to pack,inpa
```

```
     kken,to forget,vergeten,while,terwijl
1660 datasight,bezienswaardigheid,to vi
     sit,bezoeken,visit,bezoek
1670 datato spend,doorbrengen,to spend,
     uitgeven,expensive,duur,place,plaats
1680 datawithout,zonder,rest,rest,rest,
     rust,to rest,rusten,together,samen
1690 datato sail,varen,sailor,zeeman,ha
     rbour,haven,rough,ruw,to park,park
     eren
1700 datacar park,parkeerterrein,strang
     e,vreemd,stranger,vreemdeling,phot
     o,foto
1710 datacamera,fototoestel,picture,fot
     o,to emigrate,emigreren,captain,ka
     pitein
1720 datapupil,leerling,classroom,klasl
     okaal,lesson,les,teacher,leraar
1730 datato teach,leren,homework,huiswe
     rk,free,vrij,freedom,vrijheid,ques
     tion,vraag
1740 datato answer,antwoorden,answer,an
     twoord,wrong,fout,to think,denken
1750 datato spell,spellen,correct,juist
     ,to correct,corrigeren,to knoww,weten
1760 datato count,tellen,difficult,moei
     lijk,difficulty,moeilijkheid
1770 datato translate,vertalen,translat
     ion,vertaling,to write,schrijven
1780 datablackboard,bord,sentence,zin,e
     xept,behalve,exeption,uitzondering
1790 dataclever,knap,examination,examen
     ,to study,studeren,mark,cijfer,gla
     d,blij
1800 datareport,rapport,language,taal,t
     o learn,leren,to gues,raden
1810 datasubject,vak,mathematics,wiskun
     de,to seem,lijken,circle,cirkel
1820 datato draw,tekenen,drawing,tekeni
     ng,line,lijn,straight,recht
1830 datahistory,geschiedenis,past,verl
     eden,exercise,oefening,to mean,bet
     ekenen
1840 datato explain,uitleggen,sure,zeke
     r,to finish,eindigen
1850 datato read,lezen,chapter,hoofdstu
     k,page,bladzijde,boring,saai
1860 datamagazine,tijdschrift,library,b
     ibliotheek,newspaper,krant
1870 dataadvertisement,advertentie,easy
     ,gemakkelijk,story,verhaal,ghost,s
     pook
1880 datato believe,geloven,to interest
     ,interesseren,intresting,interessant
1890 datadictionary,woordenboek,to use,
     gebruiken,useful,nuttig,line,regel
1900 dataspectacles,brill,glasses,bril,
     letter,brief,ink,inkt,envelope,env
     elop
1910 datato reply,antwoorden,to receive
     ,ontvangen,stamp,postzegel,parcel,
     pakje
1920 datato post,posten,post,post,adres
     s,adres,to copy,overschrijven,may,
     mogen
1930 datato copy,kopieren,article,artik
     el,writer,schrijver,person,persoon
1940 datapersonal,persoonlijk,pencil,po
     tlood,right,rechts,drop,druppel
1950 datato speak,spreken,loud,luid,voi
     ce,stem,sound,geluid,to talk,praten
1960 dataconversation,gesprek,to listen
     ,luisteren,to phone,opbellen
1970 datato repeat,herhalen,perfect,vol
     maakt,to understand,verstaan,matte
     r,zaak
1980 datato whisper,fluisteren,to discu
     ss,bespreken,discussion,bespreking
1990 datato mention,noemen,to shout,sch
     reeuwen,silence,stilte,silent,stil
```

EINDE LISTING engels

Checksum Engels

REGEL 1	133	REGEL 250	59	REGEL 530	176	REGEL 810	10
REGEL 2	178	REGEL 260	122	REGEL 540	174	REGEL 820	157
REGEL 3	59	REGEL 270	74	REGEL 550	179	REGEL 830	55
REGEL 4	165	REGEL 280	81	REGEL 560	218	REGEL 840	244
REGEL 10	178	REGEL 290	89	REGEL 570	183	REGEL 850	35
REGEL 20	23	REGEL 300	113	REGEL 580	9	REGEL 860	9
REGEL 30	152	REGEL 310	175	REGEL 590	207	REGEL 870	191
REGEL 40	68	REGEL 320	163	REGEL 600	142	REGEL 880	14
REGEL 50	114	REGEL 330	216	REGEL 610	73	REGEL 890	0
REGEL 60	7	REGEL 340	74	REGEL 620	123	REGEL 900	37
REGEL 70	250	REGEL 350	79	REGEL 630	249	REGEL 910	134
REGEL 80	73	REGEL 360	161	REGEL 640	251	REGEL 920	25
REGEL 90	67	REGEL 370	150	REGEL 650	83	REGEL 930	246
REGEL 100	35	REGEL 380	133	REGEL 660	195	REGEL 940	5
REGEL 110	212	REGEL 390	132	REGEL 670	72	REGEL 950	16
REGEL 120	100	REGEL 400	197	REGEL 680	142	REGEL 960	148
REGEL 130	110	REGEL 410	205	REGEL 690	221	REGEL 970	226
REGEL 140	29	REGEL 420	132	REGEL 700	232	REGEL 980	184
REGEL 150	240	REGEL 430	48	REGEL 710	184	REGEL 990	124
REGEL 160	254	REGEL 440	4	REGEL 720	198	REGEL 1000	157
REGEL 170	211	REGEL 450	41	REGEL 730	116	REGEL 1010	159
REGEL 180	117	REGEL 460	254	REGEL 740	122	REGEL 1020	230
REGEL 190	39	REGEL 470	114	REGEL 750	186	REGEL 1030	226
REGEL 200	145	REGEL 480	106	REGEL 760	216	REGEL 1040	60
REGEL 210	102	REGEL 490	122	REGEL 770	21	REGEL 1050	193
REGEL 220	60	REGEL 500	181	REGEL 780	153	REGEL 1060	56
REGEL 230	120	REGEL 510	242	REGEL 790	107	REGEL 1070	191
REGEL 240	68	REGEL 520	35	REGEL 800	198	REGEL 1080	28

Vervolg checksum Engels

REGEL 1090	104	REGEL 1360	66	REGEL 1630	108	REGEL 1900	109
REGEL 1100	53	REGEL 1370	92	REGEL 1640	137	REGEL 1910	161
REGEL 1110	158	REGEL 1380	205	REGEL 1650	175	REGEL 1920	177
REGEL 1120	94	REGEL 1390	123	REGEL 1660	219	REGEL 1930	140
REGEL 1130	210	REGEL 1400	66	REGEL 1670	174	REGEL 1940	223
REGEL 1140	174	REGEL 1410	241	REGEL 1680	181	REGEL 1950	43
REGEL 1150	243	REGEL 1420	148	REGEL 1690	18	REGEL 1960	166
REGEL 1160	220	REGEL 1430	138	REGEL 1700	15	REGEL 1970	189
REGEL 1170	63	REGEL 1440	172	REGEL 1710	240	REGEL 1980	115
REGEL 1180	201	REGEL 1450	99	REGEL 1720	144	REGEL 1990	72
REGEL 1190	211	REGEL 1460	38	REGEL 1730	8		
REGEL 1200	220	REGEL 1470	78	REGEL 1740	10		
REGEL 1210	94	REGEL 1480	65	REGEL 1750	239		
REGEL 1220	253	REGEL 1490	119	REGEL 1760	158		
REGEL 1230	198	REGEL 1500	127	REGEL 1770	230		
REGEL 1240	16	REGEL 1510	154	REGEL 1780	116		
REGEL 1250	206	REGEL 1520	81	REGEL 1790	178		
REGEL 1260	244	REGEL 1530	178	REGEL 1800	237		
REGEL 1270	97	REGEL 1540	112	REGEL 1810	57		
REGEL 1280	69	REGEL 1550	32	REGEL 1820	36		
REGEL 1290	7	REGEL 1560	173	REGEL 1830	217		
REGEL 1300	28	REGEL 1570	205	REGEL 1840	30		
REGEL 1310	139	REGEL 1580	24	REGEL 1850	55		
REGEL 1320	136	REGEL 1590	59	REGEL 1860	85		
REGEL 1330	51	REGEL 1600	105	REGEL 1870	211		
REGEL 1340	219	REGEL 1610	205	REGEL 1880	13		
REGEL 1350	141	REGEL 1620	44	REGEL 1890	94		

Amiga Checksum

Deze listingtester, gemaakt door Luc de Cock uit België maakt gebruik van de RAM-disk. Het korte programma moet letterlijk worden overgetikt en worden weggeschreven met de opdracht:
SAVE"listingtester",a
Het achtervoegsel ,a is erg belangrijk, het programma wordt nu geSAVED als ASCII-file. Hierdoor kan het worden gekoppeld met een ander basicprogramma. Wilt u een ingetikt programma controleren, moet u dit op de gebruikelijke wijze inladen. De listingtester kan er nu aangekoppeld worden met de opdracht:
MERGE"listingtester
Hierna typt U in:
listtest 0,"naam" (waarbij naam de naam van het programma is). U krijgt hierbij de listing inclusief controlegetallen op het scherm.
Met de opdracht:
listtest 0,"naam" (waarbij naam de naam van het programma is). U krijgt hierbij de listing inclusief kontrole getallen op de printer.
Deze controlegetallen zijn alleen bedoeld als hulpmiddel. In 99% van de gevallen waarbij wij in de listing een ander getal hebben afgedrukt heeft u een typefout gemaakt. Het volstaat nu om deze regel(s) over te tikken en het dan opnieuw te proberen.

```
SUB ListTest(ok%,f$) STATIC '
Listingtester Amiga door Luc De Cock

' -------- Listingtester Amiga -------
' ---------- door Luc De Cock --------
' ---------- Haacht  (Belgie) --------
LIST ,"ram:"+f$
IF ok%=1 THEN OPEN "O",2,"par:" ELSE
OPEN "O",2,"scrn:"
OPEN "I",1,"ram:"+f$:t&=0
PRINT #2," Listing ";f$;" :
Checksum":PRINT #2," ":INPUT #1,a$
WHILE INSTR(UCASE$(a$),"LISTTEST")=0
IF a$="" THEN a$=CHR$(0)
c=0:FOR z=1 TO
LEN(a$):c=c+ASC(MID$(a$,z,1)):NEXT z
IF a$=CHR$(0) THEN a$=""
PRINT #2,RIGHT$("  "+STR$(c AND
255),3);"    ";
l$=LEFT$(a$,70):r$=MID$(a$,71):PRINT
#2,l$
WHILE LEN(r$)>70
l$=LEFT$(r$,70):PRINT #2,"
";l$:r$=MID$(r$,71)
WEND:IF r$<>"" THEN PRINT #2,"        ";r$
t&=t&+(c AND 255):LINE INPUT #1,a$
WEND:CLOSE 1:PRINT #2,"Totale checksum
:"+STR$(t&):CLOSE 2:KILL "ram:"+f$
END SUB
```

SCHUIFRAAM

De bedoeling van het spel is om de onderaan in beeld geplaatste letters van de zin ook op het schuifraam in de juiste volgorde te krijgen. Men kan de letters verplaatsen door de muiswijzer erop te zetten en op de linkermuisknop te drukken. Vervolgens zet men de muiswijzer op de lege plaats in het schuifraam en drukt men wederom op de linker muisknop. Opgelet: de te verplaatsen letter moet naast het lege vak liggen anders kan men niet verschuiven zowel beurten als tijd worden bijgehouden. Het spel is geschreven door Walter Frederickx uit Borgerhout in België.**LET OP: de controlegetallen moeten niet worden overgenomen, zij dienen alleen als hulpmiddel bij het overtikken. Zie ook de uitleg bij de listingtester !!**

```
  3  'CLEAR,120000&
210  SCREEN 1,325,250,5,1  :DIM a$(40)
     :DIM welk(40):DIM h(40):DIM
     v(40):DIM w(40)
245  WINDOW 1,"              ",,16,1
199  bbeurt=1000:btijd=1000
167  PALETTE 0,0,0,0:PALETTE
     3,1,.87,.73:PALETTE 1,0,0,0
217  PALETTE 5,0,0,0:PALETTE
     6,0,0,0:PALETTE 1,0,0,0
147  PALETTE 28,0,0,0:PALETTE 27,0,0,0
127  herbegin: tijd=0:beurt=0
 34  GOSUB kader :GOSUB titel:GOSUB
     zoekkader:GOSUB wleegmaken:GOSUB
     zoekkader1
 22  b=52:a=111 :GOSUB tijd
230  blokjes: WHILE a<200:
     LINE(a,b)-(a+14,b+14),29,bf
244  LINE(a,b)-(a+1
     4,b+14),2,b:a=a+16: WEND
170  IF b<120 THEN b=b+16:a=111:GOTO
     blokjes
203  GOSUB tekst
199  LINE (191,132)-(206,146),4,bf
122  schuivenmaar:w=0:a=1:antwoord$=""
 58  WHILE
     a<37:antwoord$=antwoord$+a$(a):a=a+
     1:WEND
220  IF antwoord$=tekst$ THEN
125  IF tijd<btijd THEN btijd=tijd
     :GOSUB tijd1
 89  IF beurt<bbeurt THEN bbeurt=beurt
     :GOSUB tijd1
165  IF pp=0 THEN
     pp=1:bbeurt=beurt:btijd=tijd
123  COLOR 17,29:GOSUB juist:FOR a=1
     TO 10000:NEXT:GOTO herbegin
 38  END IF
 35  LOCATE 27,3:COLOR 2,3:PRINT
     antwoord$
227  kijk:
```

```
6 MOUSE ON
224 ON MOUSE GOSUB muiskijken
4 IF w>0 AND welk(w)=0 THEN
van=w:GOTO kijk1
254 IF w>0 AND welk(w)=1 THEN GOTO
schuivenmaar
147 GOSUB zet
162 GOTO kijk
20 kijk1:
132 w=0
87 k2:  ON MOUSE GOSUB muiskijken
0 IF w>0 AND welk(w)=1 THEN
naar=w:GOTO nakijk
253 IF w>0 AND welk(w)=0 THEN GOTO
schuivenmaar
147 GOSUB zet
150 GOTO k2
178 nakijk:
199 IF welk(naar)=1 AND naar=van+1
THEN GOTO rechts
137 IF welk(naar)=1 AND naar=van-1
THEN GOTO rechts
140 IF welk(naar)=1 AND naar=van+6
THEN GOTO rechts
142 IF welk(naar)=1 AND naar=van-6
THEN GOTO rechts
63 GOTO schuivenmaar
84 rechts:GOSUB duur1
170 SWAP welk(van),welk(naar):SWAP
a$(van),a$(naar)
20  LINE(h(naar)-1,v(naar)-1)-
(h(naar)+13,v(naar)+13),30,bf
125 LINE(h(naar)-1,v(naar)-1)-(h(naar)+
13,v(naar)+13),2,b
115 LINE(h(van)-1,v(van)-1)-(h(van)+14,
v(van)+14),4,bf
62 LOCATE x*2+2,y*2+1:COLOR
2,30:PRINT PTAB(y*16+2) a$(naar)
63 GOTO schuivenmaar
43 duur1:
157 beurt=beurt+1
213 GOTO tijd1
229 tijd:
233 LINE(8,42)-(75,157),3,bf
184 LINE(8,42)-(75,157),17,b
122 LINE(238,42)-(305,157),3,bf
73 LINE(238,42)-(305,157),17,b
126 COLOR 17,3:LOCATE
7,4:PRINT"TIJD":LOCATE 7,33:PRINT
PTAB(252)"BEU
RT"
159 LOCATE 14,3:PRINT
PTAB(21)"BESTE";:PRINT
PTAB(252)"BESTE"
173 zet:
227 qq=qq+1:IF qq=80 THEN
qq=0:tijd=tijd+1:GOTO tijd1
160 RETURN
70 tijd1:COLOR 2,3
167 LOCATE 9,3:PRINT tijd:LOCATE
9,32:PRINT beurt
199 LOCATE 16,3:PRINT btijd:LOCATE
16,32:PRINT bbeurt
96 RETURN
```

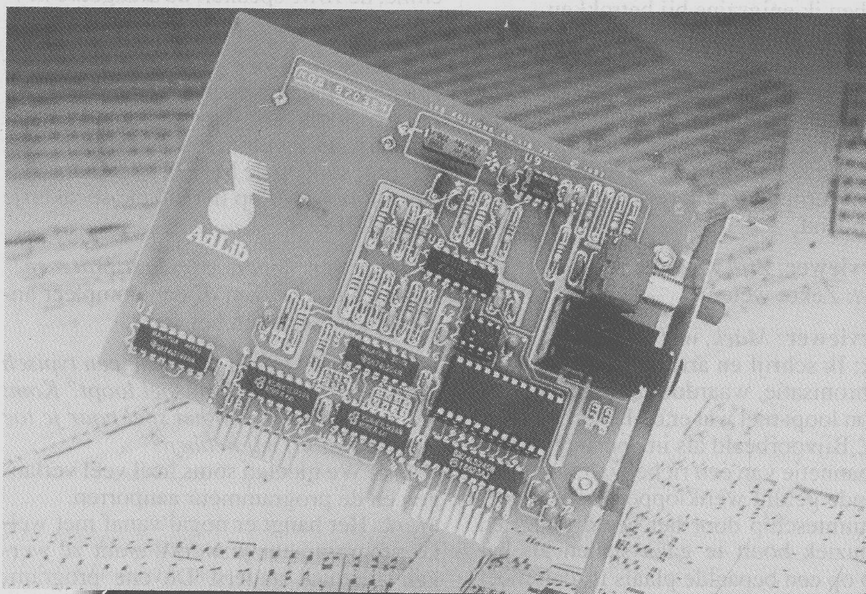
```
101 tekst:
83 READ tekst$ :READ mix$
225 lees=lees+1 :IF lees=12 THEN
lees=1:RESTORE
50 LOCATE 23,3:COLOR 17,3:PRINT
tekst$
240 a=35 :b=1
40 WHILE a>0
32 a$(b)=MID$(mix$,a,1)
20 a=a-1 :b=b+1
46 WEND
104 COLOR 2,29
163 juist: a=1:b=5:c=8:d=0
159 WHILE a<36
219 LOCATE c,b:PRINT
PTAB(115+d)a$(a):BEEP
250 FOR h=1 TO 1000:NEXT
44 d=d+16:a=a+1
211 IF d>86 THEN d=0:c=c+2:
110 WEND
160 RETURN
142 DATA --EEN ZWALUW MAAKT NOG GEEN
ZOMER--
174 DATA EEAUAAOEEOE
----NZWLWMKTNGGNZMR
14 DATA -JE BENT NOOIT TE OUD OM TE
LEREN--
24 DATA
---JBNTNTTDMTLRNEEOOIEOUOEOE
66 DATA -HOE MEER MENSEN, HOE MEER
VREUGDE-
66 DATA --OEEEEEOEEEEEU
,HMRMNSNHMRVRGD
108 DATA -GEEN SLAPENDE HONDEN WAKKER
MAKEN-
108 DATA MKNAE KKWRAE HNDNOE-
NDPLSAEE- GEEN
54 DATA ROEIEN MET DE RIEMEN DIE MEN
HEEFT-
86 DATA RIEMEN IED NEM TFEEHMET DE
-EOEI RN
34 DATA -HET ZIJN TWEE HANDEN OP EEN
BUIK--
66 DATA EIJEEAEOEEUI  HT ZN  WT
HNDN--PN-BK
51 DATA -GEDANE ZAKEN NEMEN GEEN
KEER MEER-
51 DATA EEEEEEEEEE--
GDNZKNNMNGNKRMRAA
49 DATA DE APPEL VALT NIET VER VAN
DE BOOM-
49 DATA MOOB ED NAV REV TIEN- TLAV
LEPPA DE
3 DATA -EEN GOED BEGIN IS HET HALVE
WERK--
3 DATA WRKE SHTEI HA LVE GIN
BE--GDOE- EEN
141 DATA HET BESTE PAARD STRUIKELD
WEL EENS-
141 DATA HTBST PRD-STRKLD WL NS
EEEAAUI EEEE
150 DATA WIE NIET HOREN WIL MOET MAAR
VOELEN
```

```
176 DATA VOELEN WNNN IEIE HROE WIL
    TOEM RAAM
 99 DATA --IEDER HUISJE HEEFT ZIJN
    KRUISJE--
 99 DATA -NIJZ-KRUISJE-TFEEH-EIEDR
    UIEHSJ
250 zoekkader:
237 LINE(7,169)-(305,192),3,bf
188 LINE(7,169)-(305,192),17,b
132 LINE(8,170)-(304,191),9,b
 32 RETURN
 43 zoekkader1:
220 LINE(7,201)-(305,224),3,bf
171 LINE(7,201)-(305,224),17,b
122 LINE(8,200)-(304,223),9,b
 32 RETURN
116 muiskijken:
149 IF MOUSE(0)=0 THEN RETURN
 98 y=MOUSE(3):x=MOUSE(4):l=y:y=INT(y/1
    6):x=INT((x-4)/16)
  6 IF x=3 THEN
118 IF y= 7 THEN w=1 :
    h(1)=112:v(1)=53 :RETURN
129 IF y= 8 THEN w=2 :
    h(2)=128:v(2)=53 :RETURN
131 IF y= 9 THEN w=3 :
    h(3)=144:v(3)=53 :RETURN
140 IF y=10 THEN w=4 :
    h(4)=160:v(4)=53 :RETURN
151 IF y=11 THEN w=5 :
    h(5)=176:v(5)=53 :RETURN
153 IF y=12 THEN w=6 :
    h(6)=192:v(6)=53 :RETURN
102 END IF
  7 IF x=4 THEN
175 IF y= 7 THEN w=7 :
    h(7)=112:v(7)=69  :RETURN
186 IF y= 8 THEN w=8 :
    h(8)=128:v(8)=69  :RETURN
188 IF y= 9 THEN w=9 :
    h(9)=144:v(9)=69  :RETURN
218 IF y=10 THEN w=10:
    h(10)=160:v(10)=69:RETURN
229 IF y=11 THEN w=11:
    h(11)=176:v(11)=69:RETURN
231 IF y=12 THEN w=12:
    h(12)=192:v(12)=69:RETURN
102 END IF
200 IF x=5 THEN
212 IF y= 7 THEN w=13:
    h(13)=112:v(13)=85:RETURN
223 IF y= 8 THEN w=14:
    h(14)=128:v(14)=85:RETURN
225 IF y= 9 THEN w=15:
    h(15)=144:v(15)=85:RETURN
234 IF y=10 THEN w=16:
    h(16)=160:v(16)=85:RETURN
245 IF y=11 THEN w=17:
    h(17)=176:v(17)=85:RETURN
247 IF y=12 THEN w=18:
    h(18)=192:v(18)=85:RETURN
102 END IF
201 IF x=6 THEN
 11 IF y= 7 THEN w=19:
    h(19)=112:v(19)=101:RETURN
251 IF y= 8 THEN w=20:
    h(20)=128:v(20)=101:RETURN
253 IF y= 9 THEN w=21:
    h(21)=144:v(21)=101:RETURN
  6 IF y=10 THEN w=22:
    h(22)=160:v(22)=101:RETURN
 17 IF y=11 THEN w=23:
    h(23)=176:v(23)=101:RETURN
 19 IF y=12 THEN w=24:
    h(24)=192:v(24)=101:RETURN
102 END IF
 10 IF x=7 THEN
  9 IF y= 7 THEN w=25:
    h(25)=112:v(25)=117:RETURN
 20 IF y= 8 THEN w=26:
    h(26)=128:v(26)=117:RETURN
 22 IF y= 9 THEN w=27:
    h(27)=144:v(27)=117:RETURN
 31 IF y=10 THEN w=28:
    h(28)=160:v(28)=117:RETURN
 42 IF y=11 THEN w=29:
    h(29)=176:v(29)=117:RETURN
 17 IF y=12 THEN w=30:
    h(30)=192:v(30)=117:RETURN
102 END IF
203 IF x=8 THEN
254 IF y= 7 THEN w=31:
    h(31)=112:v(31)=133:RETURN
  9 IF y= 8 THEN w=32:
    h(32)=128:v(32)=133:RETURN
 11 IF y= 9 THEN w=33:
    h(33)=144:v(33)=133:RETURN
 20 IF y=10 THEN w=34:
    h(34)=160:v(34)=133:RETURN
 31 IF y=11 THEN w=35:
    h(35)=176:v(35)=133:RETURN
 33 IF y=12 THEN w=36:
    h(36)=192:v(36)=133:RETURN
134 END IF
  0 RETURN
 41 wleegmaken:a=1
169 WHILE a<37 :welk(a)=0:a=a+1:WEND
213 welk(36)=1:RETURN
 47 kader: LINE(101,42)-(215,155),4,bf
119 LINE(101,42)-(215,155),12,b
 74 LINE(100,41)-(216,156),6,b
 80 LINE(109,50)-(207,148),2,b
160 RETURN
 92 titel:
 33  LINE(7,1)-(305,35),
    3,bf:LINE(8,2)-(304,34),9,b:LINE(7,
    1)-(305,35),17,b
201 COLOR 17,3:LOCATE 2,3
 25 PRINT" >> SCHUIFRAAM FREDERICKX
    WALTER << "
221 LOCATE 3,4:PRINT"HELMSTR 128 2200
    BORGERHOUT BELGIE"
128 RETURN
```

Totale checksum : 23756

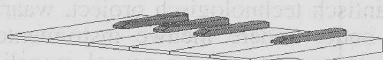
Sierra's muziekmakers

De mannen die muziek binnen een paar maanden van een enkele stem naar stereo synthesizer soundtracks brachten



De muziek in de Sierra-spelletjes heeft in de afgelopen twee jaar een enorme vlucht genomen, de enkele stem die in de meeste computers normaal is (de stem die tegen je piept) is veranderd in prachtige stereofonische muziek die via een synthesizer wordt gespeeld; de meest complexe daarvan heeft 32 kanalen.

De twee mannen die voornamelijk verantwoordelijk zijn voor deze technologische vooruitgang zijn Stuart Goldstein en Mark Siebert, Sierra's eigen professionals op het gebied van programmeren en muziek. Deze twee stille mannen zien er in het geheel niet revolutionair uit, maar samen hebben ze in de afgelopen 18 maanden wel gezorgd voor een revolutie in de muziek in geanimeerde 3D adventurespellen. Tijdens het interview lachten we om het feit dat, toen zij bij Sierra begonnen, hun plekje de muziekkamer 'werk' genoemd werd, terwijl het nu De Afdeling Muziek heet en een heuse werkeenheid is, die nu vier mensen omvat.Ik sprak met hen om uit te vinden waar ze vandaan komen, wat ze doen en waar ze naartoe gaan.



Sierra werd in 1989 in de VS door de Software Publishers Association genomineerd voor 'best achievement in sound', voor de muziek in King's Quest IV evenals voor 'overall technical achievement' voor SCI. De Excellence in Software Awards worden elk jaar door de SPA uitgereikt, waarbij net zoals bij de Oscar-uitreiking ook een nominatie al een grote eer is.

***Interviewer:** Kunnen jullie me om te beginnen in het kort iets vertellen over jullie achtergrond?*

Stuart: Ik heb geleerd voor computerprogrammeur, en daarnaast ben ik part-time musicus en heb in verschillende bands gespeeld. Ik componeer thuis veel en heb een kleine MIDI-studio. Ik kwam bij Sierra als programmeur van de software afdeling en had drivers nodig voor de muziek in de Sierra uitgaves.

Mark: Mijn achtergrond is muziek. Ik studeerde muziek en tourde daarna vijf jaar met een professionele band, waarmee ik ook een paar platen opnam en wat lokale hits had. Nu speel ik bij een band in een club. Ik wilde terug naar school gaan om wiskunde en informatica te studeren toen ik de advertentie van Sierra zag voor een MIDI-musicus. Dat was precies wat ik had gedaan, dus kwam ik hier werken.

***Interviewer:** Moet je voor dit werk zowel computerprogrammeur als professionele musicus zijn?*

Stuart: Niet allebei, maar wel elementen uit beide. Mijn achtergrond is programmeren maar de muzikale ervaring hielp. Datzelfde geldt voor Mark en in het midden is een overlapping, zodat we elkaars individuele mogelijkheden aanvullen.

Interviewer: *Hoe werken jullie als team samen?*

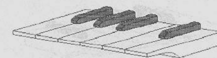
Mark: Het start meestal bij mij met een idee voor muziek. We werken bijvoorbeeld nu aan een nieuw spel, en ik had een idee voor een liedje. Ik begon eraan te werken en ontdekte wat problemen die het systeem niet aankon, ik kon sommige dingen niet die ik muzikaal wilde doen. Dus ging ik ermee naar Stuart en bespraken hoe we de driver moesten aanpassen, zodat ik wel kon doen wat ik wilde.

Stuart: Precies! We zoeken dan samen uit of het haalbaar is om een nieuwe mogelijkheid aan het systeem toe te voegen. We proberen voortdurend de grenzen van de mogelijkheden te verleggen. Soms gaat het ook andersom, als ik een idee krijg dat een verbetering betekent en dat brainstorm ik dan met Mark.

Interviewer: *Daarmee krijgen jullie veel creatieve mogelijkheden, Vertel eens hoe dat werkt.*

Mark: We doen vooral dingen met de MT-32 (synthesizer van het merk Roland, vert.) en sommige andere kaarten, die niet voor de gebruiker toegankelijk zijn en evenmin door standaard MIDI-manipulatie door een sequencer.

Stuart: We maken effecten op de MT-32 die voor de gemiddelde persoon niet mogelijk zijn.



Interviewer: *Welke verantwoordelijkheden dragen jullie bij Sierra?*
Stuart: Onze huidige systemen - onze ondersteuning van een boel geluidsopties - ontwikkelen en onderhouden. Ook moet ik uitvinden hoe we MIDI-gegevens kunnen vertalen naar een apparaat dat we besluiten te ondersteunen. We bekijken veel van de nieuwe kaarten die binnenkomen om te zien of we ze willen ondersteunen, daar ben ik enigszins bij betrokken.
Aan de horizon gloort al een aantal echt nieuwe en opwindende systemen die we in de muziekafdeling gaan ontwikkelen. Tot nu toe zijn we nog aan de oppervlakte van de muziek bezig, er komt nog veel meer. We komen met CD-kwaliteit opnames op computerspelletjes, dat wordt echt opwindend.

Interviewer: *Zijn we daar bijna aan toe?*
Stuart: Zeker weten!

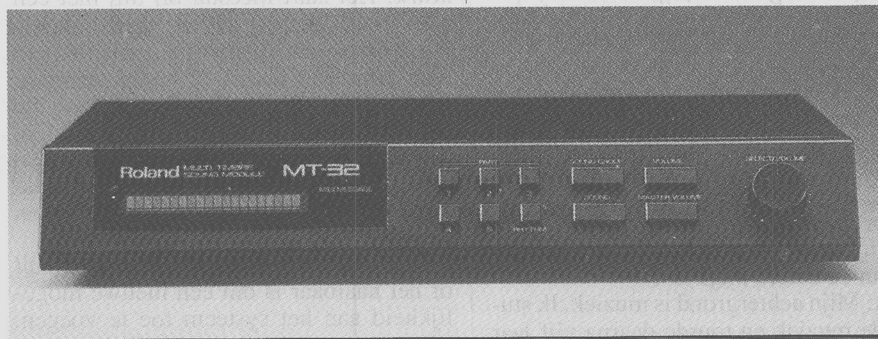
Interviewer: *Mark, wat doe jij?*
Mark: Ik schrijf en arrangeer en doe veel synchronisatie, waardoor de muziek synchroon loopt met wat er op het scherm gebeurt. Bijvoorbeeld als in *Space Quest III* het mannetje van een richel valt, moet het vallende geluid wel kloppen; of wanneer het ruimteschip door het beeld vliegt en de muziek hoort te gaan spelen als het schip op een bepaalde plaats is, dan moet alles wel op het juiste moment plaatsvinden.
We maken hier ook een heleboel geluiden zelf, dus we gebruiken geen samples. Als de MT-32 niet kan doen wat wij willen dan moeten we vanaf nul beginnen en intern de MT-32 veranderen zodat het klinkt, zoals wij willen dat het klinkt. We doen ook conversies van oudere spelletjes met muziek naar nieuwe systemen, zoals de Amiga of de Apple IIGS. We voegen muziek toe of halen juist wat weg om het met dat systeem te laten werken.
Stuart: We proberen altijd maximaal gebruik te maken van de mogelijkheden van het systeem waarvoor we schrijven. De Amiga bijvoorbeeld kan tegelijkertijd vier digitale geluidssamples terugspelen, dus daarmee proberen we iets te doen.

Interviewer: *Jullie maken dus eerst de muziek op de IBM, en als het spel ook voor een andere machine wordt overgezet converteren jullie de muziek?*
Mark: Zelfs elk IBM-spel heeft conversies in zich, want we schrijven voor de MT-32, de AdLib, de IMF, de Tandy-machine, de IBM-speaker, de draagbare keyboards van Casio, dat alles voor de IBM.

Interviewer: *Doen jullie voor elke stuk een eigen muziek-track?*
Stuart: Soms wel, we proberen zo veel mogelijk te combineren. Maar een echt sfeervol stuk op de MT-32 komt soms helemaal niet over op het enkele speakertje van de IBM.

Interviewer: *Wat doe je daar dan mee?*
Mark: We schrijven of een compleet ander stuk of we laten het weg.

Interviewer: *Vertel eens hoe een typisch muziekproject voor een spel loopt? Komt de programmeur van het spel naar je toe en zegt wat hij wil hebben?*
Stuart: We moeten soms heel veel verlangen en de programmeur aanporren.
Mark: Het hangt er nogal vanaf met welke programmeur je werkt, want ze werken allemaal anders. De ene programmeur komt naar je toe en zegt: "Dit wil ik exact zo hebben", terwijl de andere zegt: "Ik heb geen idee wat ik wil." Soms krijg je een compleet overzicht van alle details van het spel, een andere keer krijg je helemaal niets.
Stuart: Je probeert eerst gevoel te krijgen voor wat de ontwerper wil voor een bepaalde scéne en dat zet je dan op muziek.
Mark: Bijvoorbeeld bij één van de spellen waaraan ik op het moment werk zei een betrokken maker: "Het begin van *De Drie Musketiers*, zo wil ik het hebben." Dus we huurden de film en iedereen die met het spel bezig was, zowel programmeurs als artistieke makers, kwamen hier om samen de opening te bekijken. We bespraken vervolgens de dingen die we er wel en niet leuk aan vonden.
Stuart: Als je het spel ziet dan zul je dat in de muziek herkennen.
Mark: Je zult niet precies die scéne zien of die muziek horen. We proberen iets te schrijven dat juist dat emotionele beeld in de muziek overbrengt.



De Multitimbrale soundmodule MT 32

Interviewer: *Een ander nieuw spel is Roberta's moordmysterie dat speelt in het oude zuiden (van de VS, vert.). Wat hebben jullie daarmee gedaan?*
Mark: Voor dat spel deed Ken Allen veel research in de bibliotheek van Fresno.

Interviewer: *Ken, vertel me nu eerst eens hoe die research in zijn werk ging. (Ken Allen en Rob Atesalp zijn beiden 'computermusici' en werden in 1989 aan het muziekteam van Sierra toegevoegd, vert.)*
Ken: Ik begon ermee zo veel mogelijk over de plaats van handeling te weten te komen van de schrijfster van het spel, Roberta Williams. Ik ging vervolgens naar de bibliotheek en ik luisterde naar het werk van diverse artiesten uit die tijd, zoals Gershwin, George M. Cohan en Stravinsky. In die tijd was ook de muziek bij de stomme films belangrijk, en dat gaf me veel houvast. Ik kreeg daarvan een goed gevoel en besloot dan ook volgens die patronen de muziek te schrijven.

Interviewer: *Klopt het dat het de bedoeling is om historisch accuraat te zijn, om met de muziek echt in de juiste tijdssfeer te blijven?*
Stuart: Zeker, mischien kan Mark als voorbeeld vertellen over Camelot.
Mark: Camelot speelt in de negende eeuw en daarom is het puur pre-Renaissance muziek. De muziekinstrumenten werden in die tijd geheel anders gestemd, daarom gebruiken we echt de manier van stemmen uit de negende eeuw. Dat vergt een gigantisch technologisch project, waarin we onder andere nieuwe instrumenten moesten ontwerpen die zoveel mogelijk klonken als de authentieke instrumenten uit die tijd. Dat betekent wel dat je wat meer authenticiteit toevoegt aan wat je doet.

Interviewer: *Je hebt gezamenlijke projecten en je eigen creatieve ideeën, die allemaal gelijktijdig spelen. Hoe houdt je zoveel spellen creatief gezien gescheiden?*
Mark: We hebben meestal twee of drie projecten tegelijkertijd, soms zelfs meer - en dat omvat zowel muziek, software als conversies. Het management vertelt ons welk project prioriteit heeft en daar werk je dan een tijd aan.
Stuart: Je richt jezelf op dat wat er op een bepaald moment gedaan moet worden. Je denkt niet na over muziek uit de twintiger jaren als je bezig bent met middeleeuwse

muziek. Je moet het echt buitensluiten anders sluipt het toch binnen.

Interviewer: *Dit is de kamer waar ik het liefst langsloop. De geluiden - ik moet zeggen de muziek - die hier uitkomen zijn soms fascinerend. Hoe beslis je wiens muziek er gespeeld wordt?*
Mark: Dat is nog geen probleem geweest. Nu we met meer mensen zijn gaan we wel meer met koptelefoons werken.
Stuart: Je krijgt in onze groep ook heel snel feedback. Als je hoorbaar via de speakers bezig bent heb je direct twee man publiek. Men zegt dan soms: "Dat is erg goed, ik vind het leuk." Het is zelfs zo dat je weet dat het nog niet klaar is als je geen reactie krijgt.

Interviewer: *Betekent de personeelsuitbreiding van twee naar vier dat het goed gaat met de muziek in de spelletjes?*
Mark: Ja, *King's Quest IV* was het eerste grote SCI-spel, en veel spellen worden nu zo groot.
Stuart: Ook het aantal projecten en het feit dat bij alle spellen uitgebreide soundtracks horen is bepalend, mensen gaan het verwachten.

Interviewer: *Raak je wel eens gefrustreerd door technologie?*
Stuart: Nou, ik zie Mark soms wel eens wanhopig worden. Hij wil in feite een idee uitdragen maar er zijn nu eenmaal begrenzingen waarvan je je bewust moet zijn.
Mark: Op welk systeem je ook werkt, beperkingen zijn er altijd.
Stuart: Ja, totdat we audio-signalen feitelijk op een medium zoals de CD gaan opnemen.
Mark: Zelfs dan blijft de grote frustratie de vraag welk stuk je gaat spelen, hoe je dat samenstelt, en hoe je het meer op een echte film laat lijken. Dat is eigenlijk wat we doen, we willen een soundtrack voor een film maken, en je gaat niet naar een film waar het geluid opeens twintig minuten lang wegvalt. Dat soort situaties willen we ook in spelletjes vermijden, zodat er een continue soundtrack ontstaat.
Stuart: Je kunt het ook vergelijken met een stomme film. Daarbij was oorspronkelijk geen geluidsband, dus kwam er een pianist die iets speelde dat zo goed mogelijk aansloot bij de beelden. We behelpen ons op het moment door geluid toe te voegen aan het beeld, maar we verwachten een aantal ontwikkelingen waardoor synchronisatie echt mogelijk wordt en een spel er meer als een film uit zal zien.

Interviewer: *Ter vergelijking kun je de film Singing in the Rain bekijken, die gaat daarover. Het ene moment zijn de stomme films een rage, en toen opeens de gesproken films. De filmindustrie veranderde de produktie en keek niet meer achterom.*
Mark: Daar zijn wij op het moment ook. Sierra is de muziek ingegaan en de bal rolt nu gewoon.

Interviewer: *We hebben wel gelachen over de ontwikkeling van muziekkamer tot afdeling, maar de muziekkamer zelf bestaat nog maar kort. Jullie begonnen een jaar geleden met de muziekkamer.*
Stuart: Precies, tot dan toe was er de eenstemmige melodie, niet te vergelijken met wat we nu bieden. Het is natuurlijk zo dat mensen met een 'kale' machine ook nu nog alleen die kale melodieën kunnen horen. Daarmee kunnen we niet veel verder als we ook de graphics willen doen zoals die nu zijn. Op de PC kan de muziek wel iets beter, maar dat zou wel ten koste gaan van de vloeiende animatie. We moeten nu al moeite doen om de tijd te verdelen tussen muziek en graphics.

Interviewer: *Wat droom je dat de technologie als volgende zal brengen?*
Mark: We zijn nu met wat dingen bezig die een sterke vooruitgang zullen betekenen voor de muziek, als het lukt tenminste.

Interviewer: *De hele industrie moet dus gewoon vooruit blijven gaan, zoals overigens ook gebeurt?*
Stuart: Precies, er is op het moment al een dozijn interfacekaarten waarmee betere muziek mogelijk wordt.

Interviewer: *Tot een jaar geleden had ik nog niet eens gehoord van interfacekaarten. Was ik gewoon slecht geïnformeerd of zijn ze zo recent?*
Mark: Ze zijn zo recent. De MPU401 is vijf of zes jaar oud, en toen die uitkwam leek het erop dat alle hardwaremensen opeens de mogelijkheden van een kaart in de machine inzagen.

Interviewer: *Sierra gaat ook zelf muzieksoftware uitbrengen, is het niet?*
Stuart: Dat klopt, samen met een andere Sierra-programmeur, Pablo Ghenis, schrijf ik een sequencer-programma dat door Sierra uitgebracht zal worden.

Interviewer: *Leg eens uit wat een sequencer-programma is aan iemand die de terminologie niet kent.*
Stuart: In principe stelt een sequencer je in staat om muziek te schrijven en veranderen - dus liedjes te maken - en die via een synthesizer af te spelen, of zelfs via de geluidskanalen van de computer, alhoewel een computer met een enkel geluidskanaal natuurlijk beperkt is.

Interviewer: *Vertel eens over jullie sequencer?*
Stuart: We hebben ons programma ontwikkeld voor zowel de net beginnende musicus als voor de meer gevorderde gebruiker van sequencers. Het maakt gebruik van pull-down menu's en pop-up schermen en kan met de muis worden bediend, alhoewel dat niet noodzakelijk is. Je kunt met een keyboard spelen of via het grafische interface op het scherm, en dan kun je elke noot of elk accoord wijzigen. Je kunt ook bijvoorbeeld veranderingen aanbrengen in het tempo en in de toonsoort (transponeren) en de noten kunnen op de tel gezet worden (quantizeren).

Interviewer: *Bedoel je "beginnend" letterlijk, kan iemand die geen muziek kan lezen maar wel altijd songs wilde schrijven er ook mee overweg?*
Stuart: Ja, iemand zonder muziekervaring kan met dit programma leren. Ook de gevorderde gebruiker zal het programma waarderen, bijvoorbeeld vanwege de quantizeerfuncties.

Interviewer: *Wanneer kunnen we het verwachten?*
We werken nu aan de versie voor MS-DOS, die in het voorjaar van 1990 verwacht wordt. Daarna volgt de omzetting naar de Atari, Amiga en Macintosh.

Interviewer: *Het klinkt in ieder geval opwindend. Laten we teruggaan naar de muziekafdeling. Het lijkt mij dat jullie met de gecombineerde kennis en ervaring de spelletjes sterk beïnvloed hebben en veel van doen hadden met het niveau van de muziek in de spelletjes van het moment.*
Stuart: Dat klopt. Toen we hier kwamen hadden we geen illusies over de beperkingen. Ik speel zelf niet veel computerspelletjes. Ik heb natuurlijk wel eens wat gespeeld, maar ik probeer wat we hier doen daarmee niet te vergelijken en ik probeer ook geen beperkingen op te leggen aan wat er zou kunnen.
Mark: Ik heb nooit erg van computerspelletjes gehouden. Ik hoorde wel altijd dat ze leken op een film, maar de eerste paar spelletjes die wij deden leken echt niet op een film, dat waren nog steeds computerspelletjes. Er komt een mannetje op en er werd een deuntje afgedraaid, dat was alles.
Stuart: Je kunt niet zeggen dat wij hier in deze kamer werden opgesloten terwijl de rest van de wereld gewoon verder ging en wij in het duister aan het ontwikkelen waren. Het leek er echter wel een beetje op. Op een gegeven moment keken we naar buiten en bleek dat Sierra ver voorop was met muziek in software.

Interview door Nancy Smithe; overgenomen uit Sierra Newsletter summer 1989.
Vertaling: *Jan Boers*.

Een scherm van papier

Wie heeft er wel eens met afgunst naar een MS-DOS PC/XT/AT gekeken? Niemand? Nee, eigenlijk ook wel logisch want een machine als de Amiga steekt natuurlijk wel even met kop en schouders boven dergelijke lompe bakken uit. Toch is er één aspect waar wij jaloers op bleven. Het gemak waarmee een PC-bezitter even een uitdraai van z'n scherm op de printer maakt(e). Dan gingen we onszelf maar weer afvragen, waarom wij niet? we hebben toch ook zo'n leuk 'PrtSc'-knopje? Er gebeurt alleen niets als we op dat knopje drukken. Fortjoenatelie do we have our brains on the right place, yes!

Wij hadden echt behoefte aan een simpele druk op een willekeurig knopje op het toetsenbord, het PrtSc-knopje, en voilà, daar staat het geheel op papier. En Commodore zou Commodore niet zijn als ze de Amiga niet had uitgerust met een handig commando genaamd PRD_DUMPRPORT, maar hierover straks meer.

Het probleem

Het probleem was dus om een routine te schrijven welke de PrtSc-toets zou monitoren. Echter een simpele oplossing als while(key != PRTSC); was niet toegestaan daar deze onaanvaardbaar veel rekentijd in beslag zou nemen. We moesten dus een andere oplossing zoeken. Gelukkig dat er een bekende uitgeverij bestaat die de Duitse boeken van Data Becker vertaal.td naar het Nederlands. De Amiga Intern hielp ons een heel eind in de juiste richting. De correcte oplossing was om de task (lees routine) te laten wachten, door middel van de functie Wait(), totdat er een signaal opgetreden was. Dit signaal werd dan veroorzaakt door het indrukken van de PrtSc-toets. Was het dan ook maar zo gemakkelijk dat dit signaal gewoon door het systeem gegeven werd. Niet dus.

Input.device

In eerste instantie dachten we gemakkelijk af te zijn door de toetsenbordgegevens van het huidige Window af te tappen. Ha, weer een illusie armer. Ook dit ging dus niet. De toetsenbord-data kwamen netjes binnen, daar zat het probleem 'm niet. Nee, ons ingrijpen had een naar zij-effect. Toetsenbord invoer in DOS was totaal onmogelijk geworden!! Tja, daar zit je dan met je goede ideeën. Een perfect werkende routine, echter alles is voor niets omdat je geen gegevens meer achter de 'DOS prompt' in kunt voeren. Niet echt het idee van een perfect werkende PrtSc-toets handler. Ook hier kwamen de heren van Data Becker ons weer te hulp. Na een kwartiertje bladeren en lezen in de Amiga Intern 2.0 werd het ons duidelijk dat we gebruik moesten maken van het input.device. Zet een input-handler op en laat deze controleren op de bewuste toetsencombinatie. Als deze zich voorgedaan heeft stuur dan een signaal naar de wachtende taak. Deze komt dan uit zijn slaap. Deze taak opent vervolgens het printer.device, stuurt het commando PRD_DUMPRPORT naar het printer.device, wacht tot het geheel op het papier staat en sluit de boel weer om maar weer in slaap te gaan tot de volgende keer dat de PrtSc-toets ingedrukt wordt.

Van theorie naar praktijk

Het idee eenmaal in ons hoofd hebbende, begonnen we te programmeren. Het boek Amiga Intern 2.0 erbij, routines uitpluizen, vervolgens op het zeer gladde ijs van de ons nog onbekende machinetaal begeven, en het lukte!
Hoe ziet het programma eruit? We geven eerst een beschrijving per functie.
We onderscheiden een aantal verschillende soorten routines. De eerste soort zijn de input.device routines. Dit zijn achtereenvolgens, g_InputCode, de interrupt-handler welke de eigenlijke C-Handler oproept. Deze is op machinetaal-niveau geprogrammeerd daar een C-routine z'n parameters via de stack overdraagt. Het input.device doet dit niet. Deze dragt z'n parameters over via de hardwareregisters. De bedoeling was dus een conversie-routine op te zetten welke de hardware registers op de stack duwt waarmee de C-routine dan weer kan werken. Deze functie vervult g_InputCode. PH_InputRemHandler() verwijdert een toegevoegde input-handler. PH_InputAddHandler() voegt een nieuwe inputhandler toe aan de lijst. Tot slot, PH_KeyBoardHandler() is de eigenlijke routine welke wordt opgeroepen als er een inputevent, dit kan zijn muisinvoer, toetsenbordinvoer, diskactiviteit, verandering aan Windows etcetera, voorkomt.
Vervolgens onderscheiden we twee algemene device-support routines. PH_SetUpDevice() creëert de benodigde MsgPort en devicerequest structuren en opent het opgegeven device. De naam PH_KillDevice() doet al vermoeden dat de bedoeling van deze functie is om het device weer te sluiten. Tevens verwijderd het de gereserveerde devicerequest structuren en verwijderd het de Message Port. PH_PrintError() print een error boodschap op het scherm.
De laatste groep is de groep, programma-specifieke functies. Dezen zijn specifiek voor het programma bedoeld. Bijvoorbeeld, PH_OpenAll(), PH_CloseAll en PH_PrintRastPort(). Eén blik op de titel van deze routines doet meteen al vermoeden wat de werking ervan is.
Eén uitzondering op de bovenstaande functiegroepen, GP_BailOut(). De prefix staat voor General Purpose, deze routine dient om, ingeval van een noodgeval, snel uit het programma te kunnen komen. Het is echter wel een zeer grove maatregel daar het alle geopende routines, structuren etcetera open laat.

De hoofd-'loop'

Het programma draait om de volgende punten:

- 1 open het input.device, voeg de inputhandler toe en bepaal welk signaal gebruikt mag worden.
- 2 zet de oproepende taak mbv. Wait() op de Waiting list en wacht tot het signaal, welke door de inputhandler gegeven wordt, voorkomt.
- 3 print de rastport uit door middel van het commando PRD_DUMPRPORT en ga weer naar stap 2.

Bekijk nu de main functie. De functie PH_SetUpDevice() opent eerst het input.device. In de volgende regel wordt de inputhandler toegevoegt. Nu komt het programma in een eindeloze lus terecht,

immers er is een while(ALWAYS) vermeld waarbij ALWAYS de waarde 1 heeft. De uitkomst zal altijd TRUE zijn dus de lus is eindeloos. De volgende functie, Wait() zet de taak op wachten. Nu is het de taak aan PH_KeyBoardHandler() om het programma weer wakker te krijgen. Bekijk deze functie nu eens goed. Zoals u ziet bestaat het eigenlijk maar uit één onderdeel,

```
if(f_Input->ie_Class ==
   IECLASS_RAWKEY &&
   f_Input->ie_Code ==
   PRTSC &&
   f_Input->ie)Qualifier ==
   IEQUALIFIER_LSHIFT)
  Signal(g_ThisTask,g_Signal);
```

De functie Signal() wordt dus opgeroepen als [SHIFT]-[PRTSC] ingedrukt wordt. Dit haalt het programma weer uit zijn slaap waarna het scherm uitgeprint wordt.

Gebruiksaanwijzing

Het programma dient u te compileren en te linken met de Aztek C-compiler. Compile met 'cc prtscr.c +L' en link met 'ln prtscr.o -lm32 -lc32'. Als alles goed gegaan is, dus als u geen fouten gemaakt heeft, dan beschikt u nu over de uitvoerbare file. Om het programma op te starten typ dan in:
run prtscr
Nu zal de **PrcSc-toets** driver zich installeren. Een druk op de [shift-PrtSc] zal hierna voldoende zijn om een uitdraai van het actieve scherm op de printer te krijgen.

Ook telefonisch te bereiken

Om het de modembezitters onder ons iets gemakkelijker te maken hebben we het programma op een BBS gezet. De bedoeling is dat we dit voortaan met al onze programma's gaan doen. De BBS in kwestie is hier BBS-Sneek welke u kunt bereiken onder telefoonnr. 05150-25553. Inloggen met 1200 of 2400 baud werkt gegarandeerd.

Johan & Johan

Literatuur

Bleek, Jennrich & Schultz (1989), Amiga Intern 2.0, Data Becker, Bruna uitg. Utrecht, ISBN 90 229 3683 X
Dittrich, Gelfand & Schemmel (1988), Amiga Intern, Data Becker, Bruna uitg. Utrecht, ISBN 90 229 3473 X
Morimore, Eugene P. (1987), Amiga Programmer's handbook volume II, SYBEX Alameda, ISBN 0 895888 384 8

Instructies:

```
Programmanaam     : PrtScr.c
Programmeertaal   : C
Gebruikt Programma : Aztek C v3.6
Opties     : cc PrtScr.c +L
      : ln PrtScr.o -lm32 -lc32
Aanroep    : Run PrtScr

#include "functions.h"
#include "exec/types.h"
#include "exec/tasks.h"
#include "exec/errors.h"
#include "devices/input.h"
#include "devices/printer.h"
#include "intuition/intuitionbase.h"
#ifndef EXEC_INTERRUPTS_H
 #include "exec/interrupts.h"
#endif
#ifndef EXEC_DEVICES_H
 #include "exec/devices.h"
#endif
#ifndef EXEC_IO_H
 #include "exec/io.h"
#endif
#ifndef DEVICES_INPUTEVENT_H
   #include "devices/inputevent.h"
#endif

#define LOPRI_DOSERROR 5
#define HIPRI_DOSERROR 20
#define PRINTERNAME "printer.device"
#define INPUTNAME "input.device"
#define ALWAYS 1
#define PRTSC 0x5d

typedef struct IntuitionBase t_IntBase;
typedef struct MsgPort t_MsgPort;
typedef struct IOStdReq t_IOStdReq;
typedef struct IORequest t_IORequest;
typedef struct IODRPReq t_IODRPReq;
typedef struct Window t_Window;
typedef struct Screen t_Screen;
typedef struct Interrupt t_Interrupt;
typedef struct InputEvent t_InputEvent;
typedef struct Task t_Task;

t_IntBase *IntuitionBase;
t_Interrupt g_InputHandler;
t_Task *g_ThisTask;
ULONG g_UserRoutine,g_Signal;
```

```
void g_InputCode();

void GP_BailOut(f_How,f_Why)
int f_How;
char *f_Why;
{
 printf("%s\n",f_Why);
 exit(f_How);
}

void PH_CloseAll()
{
 if(IntuitionBase)
    CloseLibrary(IntuitionBase);
}

void PH_OpenAll()
{
 if(!(IntuitionBase=(t_IntBase*)
   OpenLibrary("intuition.library",0L)))
      GP_BailOut(LOPRI_DOSERROR,"No intuition.library");
}

void PH_PrintError(f_Value)
int f_Value;
{
 switch(f_Value)
 {
  case PDERR_NOTGRAPHICS:
   printf("Error - printer is not a graphicsprinter\n");
  break;

  case PDERR_BADDIMENSION:
   printf("Error - dimensions are incorrect\n");
  break;

  case PDERR_DIMENSIONOVFLOW:
   printf("Error - dimensions to large for printer\n");
  break;

  case IOERR_NOCMD:
   printf("Error - no command specified!\n");
  break;

  case IOERR_OPENFAIL:
   printf("Error - device failed to open\n");
  break;

  case IOERR_ABORTED:
    printf("Error - command aborted\n");
  break;

  case IOERR_BADLENGTH:
   printf("Error - bad length specified\n");
  break;
```

```
 default:
    printf("Error detected -
source unknown\n");
  break;
 }
}

void PH_PrintRastPort(f_PSRe-
quest,f_PSWindow)
t_IODRPReq *f_PSRequest;
t_Window *f_PSWindow;
{
 t_Screen *l_PSScreen;

      l_PSScreen=f_PSWindow-
>WScreen;    /* pointer naar
huidige  scherm */
    f_PSRequest->io_RastPort
=&l_PSScreen->RastPort;
     f_PSRequest->io_ColorMap
=l_PSScreen->ViewPort.Color
Map;
       f_PSRequest->io_Modes
=l_PSScreen->ViewPort.Modes;
 f_PSRequest->io_SrcX   =0L;
 f_PSRequest->io_SrcY   =0L;
     f_PSRequest->io_SrcWidth
=l_PSScreen->Width;
                 f_PSRequest-
>io_SrcHeight=l_PSScreen->H
eight;
     f_PSRequest->io_DestCols
=960L;
     f_PSRequest->io_DestRows
=(LONG)((960.0/(float)l_PSS
creen->Width)*(float)l_PSSc
reen->Height);
     f_PSRequest->io_Command
=PRD_DUMPRPORT; /* commando om
rastport te dumpen */
       f_PSRequest->io_Flags
=IOF_QUICK; /* geen queueing
*/
       BeginIO(f_PSRequest);
/* start command */
         WaitIO(f_PSRequest);
/* wacht op device */

 if(f_PSRequest->io_Error !=
NULL)  /* bij een error */
  PH_PrintError(f_PSRequest-
>io_Error);
}
void    PH_KillDevice(f_PSRe-
quest)
t_IORequest *f_PSRequest;
{
   CloseDevice(f_PSRequest);
/* sluit device   */
      DeletePort(f_PSRequest-
>io_Message.mn_ReplyPort);
 /* verwijder msgport */
   DeleteExtIO(f_PSRequest);
/* verwijder request */
}
```

```
APTR             *PH_SetUpDevi-
ce(f_Name,f_Unit,f_PSReques
t,f_Flags,f_Size)
UBYTE *f_Name;
APTR *f_PSRequest;
ULONG f_Unit,f_Flags,f_Size;
{
 t_MsgPort *l_DevicePort;
 APTR *CreateExtIO() /* maak
MsgPort    */
      l_DevicePort=(t_MsgPort
*)CreatePort(NULL,0L);
 /* maak device request*/
        f_PSRequest=CreateEx-
tIO(l_DevicePort,f_Size);
 /* open device     */
                if(OpenDevi-
ce(f_Name,f_Unit,f_PSReques
t,f_Flags))
    GP_BailOut(LOPRI_DOSER-
ROR,"No device to open");

 return(f_PSRequest);
}

/* ---------------------- */
/* de basisgedachte achter de
volgende vier routines zijn
*/
/* ontleend aan de voorbeelden
zoals ze besproken zijn in  */
/* het boek Amiga Intern 2.0,
Data Becker, pg. 93-104.    */
/* Mijn hartelijke dank voor
deze support!            */
/* ---------------------- */
#asm
 public _geta4
 public _g_InputCode;

_g_InputCode:
 move.l a4,-(sp)
 jsr   _geta4
 movem.l a0/a1,-(sp)   ; pa-
rameter op de stack
  move.l  _g_UserRoutine,a0 ;
c-routine aanroepen
 jsr   (a0)
 movem.l (sp)+,a0/a1
 move.l  (sp)+,a4
 rts
#endasm

void PH_InputRemHandler(f_In-
putRequest)
t_IOStdReq *f_InputRequest;
{
 FreeSignal(g_Signal);    /*
geef signaal weer vrij */
     f_InputRequest->io_Data
=(APTR)&g_InputHandler;
      f_InputRequest->io_Com-
mand=IND_REMHANDLER;
/* en verwijder handler  */
 DoIO(f_InputRequest);
}
```

```
void PH_InputAddHandler(f_In-
putRequest,f_Handler,f_Data)
t_IOStdReq *f_InputRequest;
void *f_Handler;
APTR f_Data;
{
 /* reserveer signaal voor
  tasksignalering        */
  if(!(g_Signal=AllocSignal(-
1L)))
       GP_BailOut(HIPRI_DOSER-
ROR,"No signal to allocate");
  /* geef correcte startadres
   voor inputhandler   */
            g_UserRoutine   =
(ULONG)f_Handler;
   g_InputHandler.is_Data   =
f_Data; /* waar is datageheu-
gen? */
 /* waar is interrupthandler?
*/
   g_InputHandler.is_Code   =
(void(*)())g_InputCode;
                g_InputHand-
ler.is_Node.ln_Pri  =  51; /*
prioriteit */
  f_InputRequest->io_Data  =
(APTR)&g_InputHandler;
 f_InputRequest->io_Command =
IND_ADDHANDLER;
 DoIO(f_InputRequest);    /*
installeer handler */
}

t_InputEvent    *PH_KeyBoard-
Handler(f_Input,f_Data)
t_InputEvent *f_Input;
ULONG f_Data;
{
  /* als er toetseninvoer is en
       [shift-prtsc]  ingedrukt
wordt */
 if(f_Input->ie_Class  == IE-
CLASS_RAWKEY &&
    f_Input->ie_Code   == PRTSC
&&
       f_Input->ie_Qualifier ==
IEQUALIFIER_LSHIFT)
       Signal(g_ThisTask,g_Sig-
nal);
  /* haal dan de oproepende taak
 uit Wait()            */
 return(f_Input);
}

main()
{
 t_IODRPReq *l_PSRequest;
 t_IOStdReq *l_PSInput;
 t_Window  *l_PSWindow;

 PH_OpenAll();
     l_PSWindow=IntuitionBase-
>ActiveWindow;
```

```
printf("\nPrtSc-key  driver
installed\n");
  g_ThisTask=FindTask(NULL);
/* waar is de huidige taak */
 g_ThisTask->tc_Flags=PA_SIG-
NAL;/* set SIGNAL mode      */

  l_PSInput=(t_IOStdReq    *)
/* open input.device    */
     PH_SetUpDevice(INPUTNA-
ME,0L,l_PSRequest,0L,sizeof
(t_IOStdReq));
 /* add handler ad. lijst   */
 PH_InputAddHandler(l_PSIn-
put,PH_KeyBoardHandler,NULL
);

while(ALWAYS)
 {
 Wait(g_Signal);          /*
wacht op [shift-prtsc]  */
 l_PSRequest=(t_IODRPReq *)
/* open printer device   */
  PH_SetUpDevice(PRINTERNA-
ME,0L,l_PSRequest,0L,sizeof
(t_IODRPReq));
 /* en print rastport    */
   PH_PrintRastPort(l_PSRe-
quest,l_PSWindow);
 PH_KillDevice(l_PSRequest);
/* verwijder device      */
 }
```

```
/* hier komt de routine nooit,
echter als iemand de conditie
  ALWAYS in while vervangt voor
een routine welke controleert
  op een bepaalde toetsencombi-
natie, bijv. ESC-c, voorkomt
  om de driver te verwijderen,
dan zullen onderstaande regels
   wel nodig zijn om de input-
handler te verwijderen en het
  input-device te verwijderen.
*/

   PH_InputRemHandler(l_PSIn-
put);
PH_KillDevice(l_PSInput);
PH_CloseAll();
}
```

Amiga C-cursus

In de vorige aflevering, Commodore Info nr.8 1989, hebben we de Windows behandeld. Een logisch vervolg hierop is het scherm, ook wel Screens genoemd. Voordat we hiermee beginnen, eerst een kleine terugblik alswel vooruitblik op IntuitionBase.

IntuitionBase

De vorige keer zijn we er al oppervlakkig op in gegaan. Deze structuur heeft u nodig als u echt serieus met Intuition wilt werken. Heeft u zich bijvoorbeeld wel eens afgevraagd hoe een programmeur er achter komt wat op een bepaald moment het actieve scherm is en waar deze in het geheugen staat? IntuitionBase levert hierop het antwoord. Enige kennis van de IntuitionBase-structuur is hier wel voor nodig.
Figuur 1 bevat de voor nu relevante IntuitionBase-structuur. Waarvoor dienen de verschillende elementen? We beginnen met de eerste, het element 'LibNode'. Deze is van het type **struct Library**. Hierin zijn de eigenschappen van de library 'intuition.library' opgenomen. Wilt u bijvoorbeeld het versienummer van de intuition.library weten, dan zou dit op de volgende manier kunnen gebeuren:

```
VersieNr=IntuitionBase->LibNode
    .lib_Version;
```

Het tweede element is de pointer naar de Viewstructuur van het eerste scherm. Dit eerste scherm zal over het algemeen het WorkBench-scherm zijn. De volgende drie elementen bevatten achtereenvolgens pointers naar de actuele Window-structuur, de actuele Scherm-structuur en de Schermstructuur van het eerste Scherm. Dit waren voor nu even de belangrijkste elementen. De overige elementen bevatten onder andere muiscoördinaten en een aantal gereserveerde elementen die u pas uit kunt lezen als u de regel

```
#define intuitionprivate 1
```

in uw programma opneemt.

Wat heb ik nou aan het bovenstaande, zult u zich afvragen. Weer even terug naar het eerste voorbeeld. Stel u wilt gegevens schrijven in het actuele Window. Dan zult u een pointer naar dit Window moeten hebben, anders zal uw poging op niets uitlopen. Heeft u het Window zelf geopend, dan zullen er geen problemen zijn. Immers de functie OpenWindow() geeft het adres naar het zo net geopende Window terug. Maar o wee als het het WorkBench-window betreft. Hiervan heeft u geen pointer. Hiervoor heeft Intuition gezorgd. Lees het element **ActiveWindow** in IntuitionBase uit en u hebt de pointer naar de Window-structuur van het actuele window. Een voorbeeld.

```
ActueleWindow =
    IntuitionBase->ActiveWindow
    ;
```

U heeft nu de pointer naar het aangeklikte window, of beter gezegd, u heeft de pointer naar de Window-structuur die bij het window behoort.

Screens of schermen

Nu u het nodige van de IntuitionBase-structuur afweet, wordt het tijd dat we verder gaan met het eigenlijke onderwerp van deze aflevering, de schermen. Een scherm is de ondergrond voor alle bewerkingen die met het grafische systeem plaats kunnen vinden. De kleuren, de graphics, tekst, etcetera worden verenigd in het scherm.
Hoe wordt een scherm gedefinieerd en geopend. In de eerste plaats zult u alle gegevens waaraan het scherm moet voldoen, de x- en y- coördinaten, de gebruikte kleuren, de videomode, enzovoort in een structuur samen moeten voegen. Hiervoor is een structuur, NewScreen genaamd, beschikbaar. Figuur 2 illustreert deze structure. De eerste zeven elementen zijn analoog aan hun tegenhangers in de NewWindow structuur. Ze bepalen de x- en y- coördinaten, de breedte en hoogte, en de keuze van de voor- en achtergrondkleuren. Het element 'depth' vormt een uitzondering. Ze komt niet voor in de NewWindow-structuur. Ze bepaalt het aantal 'bitplanes' dat het scherm nodig heeft. Het aantal **bitplanes** geeft aan met hoeveel kleuren het scherm moet werken. Dit aantal kleuren berekent u door uit te gaan van de volgende formule;

```
aant_kleuren = 2 ^ depth
```

Dus als we een scherm met een depth van 4 zouden openen, dan zou dit resulteren in een scherm met $2^4 = 16$ kleuren.
Nog een opmerking omtrent het element 'leftedge'. In de huidige KickStart-versies (1.1, 1.2 of 1.3) heeft deze nog geen betekenis. Het element is opgenomen om compatibiliteit te houden met nieuwere KickStart-versies welke wel een horizontale schermverschuiving ondersteunen.
Het volgende onbekende element **ViewModes** bepaalt wat voor schermmode er gebruikt wordt. Het kader geeft hierover meer uitleg. **Type** bepaalt een aantal eigenschappen van het scherm los van het ViewModes-element. We hebben de verschillende flags even voor u op een rijtje gezet.

- WBENCHSCREEN: een scherm dat alleen door het systeem geopend kan worden.
- CUSTOMSCREEN: een door u zelf gedefinieerd scherm.
- CUSTOMBITMAP: u heeft zelf geheugen gereserveerd voor een eigen BitMap.
- SCREENBEHIND: het scherm wordt achter alle aanwezige schermen geplaatst.
- SCREENQUIET: er worden geen gadgets getekend in het scherm.
- SHOWTITLE: er wordt geen kopregel getekend in het scherm. Voorkom dan ook dat met behulp van de rechter muisknop de balk weer getekend wordt. Hiervoor is de IDCMP-Flag RMBTRAP beschikbaar.
- BEEPING: het scherm knippert.

Het element **Font** dient een pointer naar een TextAttr-structuur te bevatten als u een nieuw font wilt gebruiken voor het scherm of het dient een NULL te bevatten als u het standaard font wilt gebruiken. Dit kan topaz.font 8 pnt of topaz.font 9 pnt zijn afhankelijk van uw keuze in de Preferences. In deze aflevering komt de TextAttr-structuur nog niet ter sprake, dit zal pas in de volgende aflevering gebeuren. Zet daarom nu eerst de pointer maar op NULL.

DefaultTitle spreekt eigenlijk voor zich. Het dient een pointer naar een tekststring te bevatten. Wilt u geen tekst in de kopregel, zet dit element dan op NULL. De laatste twee elementen zijn mogelijkheden voor meer gevorderde gebruikers. U kunt ze daarom beter nu nog niet gebruiken. In één van de volgende afleveringen zullen we behandelen hoe we Gadgets moeten gebruiken in Intuition-schermen, dus volstaan we nu met de opmerking dat het element **Gadgets** een pointer dient te bevatten naar het eerste Gadget dat u in het scherm wilt gebruiken. We dienen hierbij nog wel een opmerking te maken: het element is opgenomen voor compatibiliteit met de latere KickStart-versies. Invullen van dit element heeft nu nog geen effect.
CustomBitMap ten slotte, is een wat lastiger item. Als u de flag CUSTOMBITMAP zet in het element **Type**, dan kunt u hier de pointer naar de door u zelf gedefinieerde **BitMap**-structuur invullen.
Voor de meer geïnteresseerden, programmavoorbeeld 3 maakt gebruik van een Custom BitMap.
Het openen van een scherm geschiedt door gebruikmaking van de functie OpenScreen(). Zorg dat 'intuition.library' geopend is. Roep dan de functie aan met gebruikmaking van de volgende syntax:

```
pointer_naar_scherm =
    OpenScreen(pointer_naar_new
    screen_structuur)
```

Zoals u ziet, krijgt u een pointer naar een Screen-structuur terug.
Het sluiten van een scherm is al even eenvoudig. De functie CloseScreen() vereist een pointer naar het te sluiten scherm. De syntax:

```
CloseScreen(pointer_naar_scherm
    _structuur)
```

Zorg er dan wel voor dat u geen windows meer in het scherm open hebt staan. Is dit wel het geval dan zal een 'guru-meditatie' uw deel zijn.

Schermstructuur

Om wijzigingen aan te brengen in al bestaande schermen, hebben we een Screen-structuur nodig. Voordat we verder gaan, eerst een kleine onderbreking. Waar vinden we al deze structuren? Ze zijn van te voren, door de makers van de compiler, opgezet. U kunt ze vinden in de 'include'-directory op uw compilerschijf. Bijvoorbeeld, de Screen-structure. Deze vind u in het include bestand 'include/intuition/screens.h'. We zetten alle tot nu toe behandelde structuren op een rijtje en vermelden er bij in welk include-bestand ze te vinden zijn. Dit geldt overigens alleen voor de Aztek C-compiler daar deze onze leidraad is door deze cursus heen.

```
struct IntuitionBase:
    intuition/intuitionbase.h
struct NewScreen:
    intuition/screens.h
struct NewWindow:
    intuition/intuition.h
struct Screen:
    intuition/screens.h
struct Window:
    intuition/intuition.h
```

Gebruik het commando '#include' om het include-bestand bij uw source te betrekken en u kunt alle in het bestand gedefinieerde zaken in uw eigen programma gebruiken. Het blijft een schitterende taal, C! Goed, terug naar de Screen-structuur. Zoals u in figuur 3 al kunt zien, is de Screen-structuur nogal wat groter dan de NewScreen-structuur. Het bevat pointers naar alle onderdelen van het grafische besturingssysteem. Zo gauw u een pointer van een scherm heeft, dan heeft u ook het gehele grafische systeem in uw handen. Wat doet welk element in de Screen structuur. We beginnen weer van bovenaf.

NextScreen: dit is de pointer naar het volgende scherm.
FirstWindow: de pointer naar het eerste Window in het scherm.
LeftEdge & TopEdge: de linkerbovenhoek coördinaten. (LeftEdge heeft geen betekenis in WB1.1, 1.2 en 1.3).
Width & Height: de breedte en hoogte van het scherm.
MouseY & MouseX: de coördinaten van de muis ten opzichte van het scherm.
Flags: dit zijn dezelfde flags die we ook al bij NewScreen besproken hebben.
Title & DefaultTitle: hierin staan de pointers naar resp. de koptekst door de gebruiker ingevuld en de standaardkoptekst.
Fonts: een pointer naar een TextAttr-structuur welke het actuele font beheert.
ViewPort: dit is geen pointer, de gehele ViewPort-structuur is binnen de Screen-structuur opgenomen. Hier komen we later, in een van de volgende delen, op terug.
RastPort: ook de rastport is binnen de Screen-structuur opgenomen. Ook hierop komen we later nog terug.
BitMap: idem als de voorgaande twee, geen pointer maar de gehele structuur is binnen de Screen-structuur opgenomen. Ook hierop komen we nog terug.
LayerInfo: ook de Layer_Info-structuur is opgenomen binnen de Screen-structuur. Ze (de Layers) vormen een belangrijk onderdeel bij het beheren van de Windows. Weer een herhaling van wat we al een keer eerder hebben gezegd: we komen hierop later weer terug.

Figuur 1. De IntuitionBase-structuur

```
struct IntuitionBase {
   struct Library LibNode;
   struct View ViewLord;
   struct Window *ActiveWindow;
   struct Screen *ActiveScreen;
   struct Screen *FirstScreen;
   ULONG Flags;
   WORD MouseY,MouseX;
   ULONG Seconds;
   ULONG Micros;
   #ifdef INTUITIONPRIVATE
     (....)
   #endif
};
```

Figuur 2. De NewScreen-structuur

```
struct NewScreen {
   SHORT LeftEdge;
   SHORT TopEdge;
   SHORT Width;
   SHORT Height;
   SHORT Depth;
   UBYTE DetailPen;
   UBYTE BlockPen;
   USHORT ViewModes;
   USHORT Type;
   struct TextAttr *Font;
   UBYTE *DefaultTitle;
   struct Gadget *Gadgets;
   struct BitMap *CustomBitMap;
};
```

Figuur 3. De Screen-structuur

```
struct Screen {
   struct Screen *NextScreen;
   struct Window *FirstWindow;
   SHORT LeftEdge,TopEdge;
   SHORT Width,Height;
   SHORT MouseX,MouseY;
   USHORT Flags;
   UBYTE *Title;
   UBYTE *DefaultTitle;
   BYTE BarHeight,BarVBorder,BarHBorder;
   BYTE MenuVBorder,MenuHBorder;
   BYTE WBorTop,WBorLeft,WBorRight,WBorBottom;
   struct TextAttr *Font
   struct ViewPort ViewPort;
   struct RastPort RastPort;
   struct BitMap BitMap;
   struct Layer_Info LayerInfo;
   struct Gadget *FirstGadget;
   UBYTE DetailPen,BlockPen;
   USHORT SaveColor0;
   struct Layer *BarLayer;
   UBYTE *ExtData,*UserData;
};
```

Programma voorbeeld #1. Simpel scherm openen

```
#include "functions.h"
#include "exec/types.h"
#include
    "intuition/intuition.h"
typedef struct IntuitionBase
    T_IntuitionBase;
typedef struct Screen
    T_Screen;
typedef struct NewScreen
    T_NewScreen;
T_IntuitionBase
    *IntuitionBase;
T_NewScreen G_DemoNS = {
    0,150,640,256,4,
    0x00,0x01,HIRES,CUSTOM
    SCREEN,
    NULL,(UBYTE *)"C_Demo
    screen #1",
    NULL,NULL };

void CD_BailOut(F_How,F_Why)
char *F_Why;
int F_How;
{
 printf("%s\n",F_Why);
 exit(F_How);
}

T_Screen *CD_OpenAll()
{
 T_Screen *L_DemoScreen;
if(!(IntuitionBase=(T_Intuiti
    Base *)      /* open
    library */
OpenLibrary("intuition.
    library",0L)))
        CD_BailOut(5,"No
    intuition.library");
 if(!(L_DemoScreen=(T_Screen
    *)              /* open
    screen  */
    OpenScreen(&G_DemoNS)))
        CD_BailOut(5,"Open
    screen failure");
 return(L_DemoScreen);
}
void CD_WaitForCounter()
{
 register int i;
 for(i=0;i<1000000;i++);
}
void
    CD_CloseAll(F_DemoScreen)
T_Screen *F_DemoScreen;
{
 if(F_DemoScreen)
  CloseScreen(F_DemoScreen);
 if(IntuitionBase)
  CloseLibrary(IntuitionBase);
}

main()
{
 T_Screen *L_DemoScreen;

 L_DemoScreen=CD_OpenAll();
 CD_WaitForCounter();
 CD_CloseAll(L_DemoScreen);
}
```

Programma voorbeeld 2

```
#include "math.h"
#include "functions.h"
#include "exec/types.h"
#include "intuition/intuition-
base.h"

typedef  struct  IntuitionBase
T_IntBase;
typedef struct Screen T_Screen;
typedef      struct   NewScreen
T_NewScreen;

T_IntBase *IntuitionBase;
T_NewScreen G_DemoNS = {
   0,0,640,256,2,0x00,0x01,HI-
RES
  CUSTOMSCREEN,NULL,
   (UBYTE *)"C_Demo screen #2",
   NULL,NULL };

void CD_BailOut(F_How,F_Why)
int F_How;
char *F_Why;
{
 printf("%s\n",F_Why);
 exit(F_How);
}

T_Screen *CD_OpenAll()
{
 T_Screen *L_DemoScreen;

 if(!(IntuitionBase=(T_IntBase
*)
     OpenLibrary("intuition.li-
brary",0L)))
          CD_BailOut(5,"No intui-
tion.library");

    if(!(L_DemoScreen=(T_Screen
*)
  OpenScreen(&G_DemoNS)))
       CD_BailOut(5,"Screen open
failure");

 return(L_DemoScreen);
}

void CD_CloseAll(F_DemoScreen)
T_Screen *F_DemoScreen;
{
 if(F_DemoScreen)
  CloseScreen(F_DemoScreen);
 if(IntuitionBase)
  CloseLibrary(IntuitionBase);
}

void CD_Wait(F_Counter)
int F_Counter;
{
 while(F_Counter-- > 0);
}

void CD_ShowDemo(F_DemoScreen)
T_Screen *F_DemoScreen;
{
 register int i;

 for(i=1;i<200;i++)
             MoveScreen(F_DemoS-
 creen,0,1);
 for(i=1;i<200;i++)
  MoveScreen(F_DemoScreen,0,-
 1);

 ScreenToBack(F_DemoScreen);
  printf("\n\n\nCommentaar in
 het WorkBench scherm is nu weer-
 zichtbaar.\n");
 CD_Wait(300000);
 ScreenToFront(F_DemoScreen);

 for(i=0;i<10;i++)

  DisplayBeep(F_DemoScreen);
  CD_Wait(50000);
 }

 CD_Wait(100000);
 printf("\n\nDeze tekst wordt
 pas zichtbaar als het scherm-ge-
 sloten is.\n");
}

main()
 {
  T_Screen *L_DemoScreen;

 L_DemoScreen=CD_OpenAll();
 CD_ShowDemo(L_DemoScreen);
 CD_CloseAll(L_DemoScreen);
}
```

Programma voorbeeld 3

```
#include "functions.h"
#include "exec/types.h"
#include "exec/memory.h"
#include "graphics/gfxbase.h"
#include "intuition/intuition-
base.h"

#define LOPRI_DOSERROR 5
#define SIZE 10
#define MODIFY_RED 0x20
#define MODIFY_BLUE 0x10
#define MODIFY_GREEN 0x30

typedef  struct  IntuitionBase
T_IntBase;
typedef struct GfxBase T_GfxBa-
se;
typedef     struct     NewScreen
T_NewScreen;
typedef struct Screen T_Screen;
typedef struct BitMap T_BitMap;
typedef struct Window T_Window;
typedef struct NewWindow T_New-
Window;

T_IntBase *IntuitionBase;
```

```
T_GfxBase *GfxBase;
T_NewScreen G_DemoNS = {
    0,0,320,512,6,0x00,0x01,H
AM|LACE,
      CUSTOMSCREEN|CUSTOMBIT-
MAP,NULL,
            (UBYTE      *)"HAM
Screen",NULL,NULL };
T_NewWindow G_DemoNW = {
   0,0,320,512,0x00,0x01,CLO-
SEWINDOW,
      WINDOWCLOSE|WINDOWSI-
ZING|WINDOWDEPTH|WINDOWDRAG
|SMART_REFRESH|ACTIVATE,
   NULL,NULL,(UBYTE *)"C_Info
C-cursus demo #3",
    NULL,NULL,50,50,320,512,C
USTOMSCREEN };
void CD_BailOut(F_How,F_Why)
int F_How;
char *F_Why;
{
 printf("%s\n",F_Why);
 exit(F_How);
}
void CD_OpenAllLibs()
{
 if(!(IntuitionBase=(T_IntBa-
se *)
   OpenLibrary("intuition.li-
brary",0L)))
      CD_BailOut(LOPRI_DOSER-
ROR,"No intuition.library");
 if(!(GfxBase=(T_GfxBase *)
   OpenLibrary("graphics.li-
brary",0L)))
      CD_BailOut(LOPRI_DOSER-
ROR,"No graphics.library");
}
void            CD_Close-
All(F_MScreen,F_MWindow,F_M
BitMap)
T_Screen *F_MScreen;
T_Window *F_MWindow;
T_BitMap *F_MBitMap;
{
 register int i;
 if(F_MWindow)
   CloseWindow(F_MWindow);
/* sluit window */
 if(F_MScreen)
   CloseScreen(F_MScreen);
/* sluit scherm */
/* geef geheugen van BitMaps
terug */
 if(F_MBitMap)
        for(i=NULL;  i<G_De-
moNS.Depth; i++)
   FreeRaster(F_MBitMap->Pla-
nes[i],G_DemoNS.Width,G_Dem
oNS.Height);
 if(GfxBase)
     CloseLibrary(GfxBase);
/* sluit graphics.library */
 if(IntuitionBase)
     CloseLibrary(IntuitionBa-
se);   /* sluit intuition.li-
brary */
}
T_Screen       *CD_OpenCustomS-
creen(F_MBitMap)
T_BitMap *F_MBitMap;
{
 T_Screen *L_MScreen;
                        USHORT
L_Depth,L_Height,L_Width;
 register int i;
 L_Depth=G_DemoNS.Depth;
 L_Height=G_DemoNS.Height;
 L_Width=G_DemoNS.Width;
  /* initialiseer een BitMap
structuur */
        InitBitMap(F_MBit-
Map,L_Depth,L_Width,L_Heigh
t);
        G_DemoNS.CustomBit-
Map=F_MBitMap;/* geef bitmap
pointers door aan   screen en
window */
 G_DemoNW.BitMap=F_MBitMap;
 for(i=NULL; i<L_Depth; i++)
/*  reserveer  geheugen  voor
bitmaps */
  {
          if((F_MBitMap->Pla-
nes[i]=(PLANEPTR)
                   AllocRas-
ter(L_Width,L_Height))==NUL
L)
      CD_BailOut(LOPRI_DOSER-
ROR,"No memory for planes");
 /* clear bitmaps */
      BltClear(F_MBitMap->Pla-
nes[i],RASSIZE(L_Width,L_He
ight));
  }
 if(!(L_MScreen=(T_Screen *)
  OpenScreen(&G_DemoNS)))
      CD_BailOut(LOPRI_DOSER-
ROR,"Screen open failure");
 /* geef schermpointer door aan
window */
 G_DemoNW.Screen=L_MScreen;
 return(L_MScreen);
}
/*
```

Onderstaande subroutine maakt veelvuldig gebruik van routines en principes die tot nu toe nog niet in de cursus behandeld zijn. Neem daarom de routines alleen voor kennisgeving aan. Het is geen schande als u het niet begrijpt.

```
 */
void CD_ShowDemo(F_MScreen)
T_Screen *F_MScreen;
{
 register int x,y;
          SetRGB4(&F_MScreen-
>ViewPort,3,0,0,0);
   for(y=20;    y<(15*SIZE);
y+SIZE)
  {
     SetAPen(&F_MScreen->Rast-
Port,3); /* pixel 18 = ZWART */
            Move(&F_MScreen->Rast-
Port,18,y);
 Draw(&F_MScreen->RastPort,1
 8,y+SIZE);
  /* modificeer alleen de rode
kleur, zet de eerste pixels op
rood, 0 */
    SetAPen(&F_MScreen->Rast-
Port,MODIFY_RED+(y-20)/SIZE
);
          Move(&F_MScreen->Rast-
Port,19,y);
  /* positioneer op 19,y en trek
een lijn naar 19, y+SIZE */
   Draw(&F_MScreen->RastPort,
 19,y+SIZE);
 for(x=20;          x<(15*SIZE);
x+=SIZE)
 {
 /* modificeer alleen de groene
kleur, 0 */
 SetAPen(&F_MScreen->RastPor
t,MODIFY_GREEN+(x-20)/SIZE);
 /* plaats een rechthoek op po-
sitie  x,y  met  afmetingen
SIZE,SIZE*/
 RectFill(&F_MScreen->RastPo
rt,x,y,x+SIZE,y+SIZE);
  }
 }
}
main()
{
 register int i;
 T_BitMap L_MBitMap;
 T_Screen *L_MScreen;
 T_Window *L_MWindow;
 CD_OpenAllLibs();
   L_MScreen=CD_OpenCustomS-
creen(&L_MBitMap);
 if(!(L_MWindow=(T_Window *)
   OpenWindow(&G_DemoNW)))
       CD_BailOut(LOPRI_DOSER-
ROR,"Window open failure");
 CD_ShowDemo(L_MScreen);
 for(i=NULL; i<1000000; i++);
                   CD_Close-
All(L_MScreen,L_MWindow,&L_
MBitMap);
}
```

FirstGadget: een pointer voor compatibiliteit met de latere KickStart-versies. Het dient een pointer naar een Gadget-structuur te bevatten als men automatisch gadgets in het scherm wil.
DetailPen & BlockPen: de voor- en achtergrondkleur registers.
SaveColor0: als het scherm knippert, dan wordt in deze variabele de originele kleur opgeslagen.

Een aantal elementen hebben we niet genoemd. Ze zijn voor nu niet belangrijk daar ze alleen door gevorderde gebruikers kunnen cq zullen worden gebruikt.
Nu we bekend zijn met het reilen en zeilen van het gehele schermgebeuren wordt het tijd voor een voorbeeld.
Het eerste voorbeeld:
Programma 1 opent een scherm, telt van 0 tot miljoen en sluit het scherm weer.
Zoals u ziet, de vereiste werkzaamheden die nodig zijn voordat u het scherm kunt openen bestaan uit vier onderdelen, te weten:

- - includen van de benodigde structures uit het bestand 'intuition/intuition.h';
- - invullen van een NewScreen structure;
- - openen van de library 'intuition.library';
- - openen van het scherm mbv. de functie OpenScreen().

Na deze vier werkzaamheden zal het scherm geopend zijn.

Schermen en hun functies

Net als we bij de Windows al gezien hebben, kent het scherm een aantal functies. De functies zijn grofweg op te delen in twee soorten. Functies die eigenschappen van Screens veranderen en de functies welke gehele Screens beïnvloeden. Deze laatstgenoemde functies zijn de programma-tegenhangers van de verschillende muisfuncties die u met het scherm uit kunt voeren, bijvoorbeeld het naar beneden trekken van het scherm.
Welke functies zijn er zoal. Een alfhabetische opsomming.

- CloseScreen(pointer_naar_scherm): sluit het scherm.
- Voorbeeld: CloseScreen(scherm1)
- Parameter: struct Screen *scherm1
- CloseWorkBench(): sluit de Workbench. Dit kan alleen als u geen werkzaamheden in de Workbench hoeft uit te voeren. Pas hier wel mee op want het programma kan nergens meer op terug vallen. U trekt de vaste grond in feite onder het programma vandaan.

Voorbeeld:

```
error=CloseWorkBench();
Parameter: BOOL error;
DisplayBeep(pointer_naar_scherm
    ):
```

het scherm licht op.

Voorbeeld:

```
DisplayBeep(scherm1);
Parameter: struct Screen
    *scherm1;
MoveScreen(pointer_naar_scherm,
    delta_x, delta_y):
```

het scherm wordt een delta_x pixels in x-richting verschoven en delta_y pixels in y-richting verschoven. Opmerking: in de huidige WorkBench-versie is het verschuiven in x-richting nog niet mogelijk.

Voorbeeld:

```
MoveScreen(scherm1,dx,dy);
Parameter: struct Screen
    *scherm1;
SHORT dx,dy;
OpenScreen(pointer_naar_NewScre
    en_structuur): open een
    scherm met de in de
    NewScreen opgegeven
    eigenschappen.
```

Voorbeeld:

```
error=OpenScreen(scherm1);
Parameter: BOOL error;
           struct Screen
    *scherm1;
```

```
OpenWorkBench():
```

het WorkBench-scherm wordt geopend.

Voorbeeld:

```
scherm_WB=OpenWorkBench();
Parameter: struct Screen
    *scherm_WB;
```

```
ScreenToBack(pointer_naar_scher
    m):
```

het opgegeven scherm verdwijnt naar de achtergrond.

Voorbeeld:

```
ScreenToBack(scherm1);
Parameter: struct Screen
    *scherm1;
```

```
ScreenToFront(pointer_naar_sche
    rm):
```

het opgegeven scherm wordt het voorste scherm.

Voorbeeld:

```
ScreenToFront(scherm1);
Parameter: struct Screen
    *scherm1;
ShowTitle(pointer_naar_scherm,
    achter_of_voorgrond):
```

de schermtitel wordt zichtbaar gemaakt als deze afgedekt wordt door een BACKDROP-window.

Voorbeeld:

```
ShowTitle(scherm1,Display);
Parameter: struct Screen
    *scherm1;
            BOOL Display;
```

Opmerking:

```
Display == TRUE
```

als titelbalk op de voorgrond moet komen, bij FALSE komt deze op de achtergrond.

```
WBenchToBack():
```

plaatst het WorkBench-scherm op de achtergrond.

Voorbeeld:

```
WBenchToBack();
```

```
WBenchToFront():
```

het WorkBench-scherm wordt voor alle andere schermen geplaatst.

Voorbeeld:

```
WBenchToFront();
```

Nu alle functies behandeld zijn kan het verhelderend werken om een praktijkdemo te zien. Lees programma 2 door, type deze in en let op wat het doet.

Zoals u ziet wijkt voorbeeld 2 niet zoveel af van voorbeeld 1. Het wezenlijke verschil zit 'm grotendeels in de functie CD_ShowDemo(). Deze functie laat, zoals de naam al doet vermoeden, een kleine demonstratie zien. Het verschuift het scherm, laat het naar de achtergrond verdwijnen, om het vervolgens weer naar de voorgrond te halen en tot slot gebruikt het de functie DisplayBeep() om duidelijk te maken dat hiermee de aandacht van de gebruiker gevraagd kan worden. U ziet dus, de schermfuncties zijn zeer simpel te gebruiken. Het enige wat ze verlangen is, over het algemeen, een pointer naar de schermstructuur van het scherm dat u wilt manipuleren.

Windows in schermen

De vorige keer hebben we de window-story verteld. Het ging hier grotendeels om windows in het Workbench-scherm. Hoe krijgen we nu een window in een zelf gedefinieerd scherm?
Als u goed opgelet heeft, pak voor de zekerheid de vorige aflevering er nog maar even bij, heeft u in de NewWindow-structuur een pointer naar het scherm gevonden waar het window in moet verschij-

nen. In deze pointer, Screen genaamd, vult u het adres van de screen-structuur in van het scherm waarin u het window wilt laten verschijnen. Een voorbeeld:

```
if(!(MijnScherm=(struct
    Screen *)
  OpenScreen(&DemoScherm)))
    exit(FALSE);

DemoWindow.Screen=MijnScherm;
if(!(MijnWindow=(struct
    Window *)
  OpenWindo(&DemoWindow)))
    exit(FALSE);
```

Hierbij zijn de structuren DemoScherm en DemoWindow weer van tevoren gedefinieerd en ingevuld en zijn MijnScherm en MijnWindow als volgt gedefinieerd:

```
struct Window *MijnWindow;
struct Screen *MijnScherm;
```

In **DemoWindow.Screen** moet dus het adres naar de betreffende screen-structuur ingevuld worden.
Als u een programma schrijft dat bovenstaand verhaal volgt, zal het window in het door u zelf gedefinieerde scherm tevoorschijn komen.

Het laatste voorbeeld

Als u het gehele verhaal goed begrepen hebt, bent u in staat om zelfstandig een scherm te definiëren en te openen. Om alle theorie nog een keer in de praktijk te demonstreren en om uw kennis op de proef te stellen, hebben we voorbeeld 3 toegevoegd aan dit verhaal. Het voorbeeld maakt gebruik van een Custom BitMap en roept enkele functies aan die u nog niet eerder gezien heeft. Een hiervan is Alloc-Raster(). Het neemt geheugen weg van wat voor het systeem beschikbaar is waarvan de grootte afhankelijk is van de opgegeven rastermaten. U kunt dit geheugen gebruiken zonder dat u last krijgt van andere programma's (lees: tasks) die ook dit geheugen willen gebruiken. Meer hierover zullen we niet vertellen daar dit onderdeel in deze cursus nu nog niet aan bod komt. Het programma, wat doet het en waarom? Het antwoord op het eerste. Het opent een scherm met een custombitmap. Vervolgens opent het een window in dit scherm.
Zoals u ziet, G_DemoNW.Screen bevat de pointer naar het scherm waarin het window tevoorschijn moet komen. Nu komt het leuke deel. Het scherm dat we geopend hebben is een Hold And Modify-scherm, dus wel even mooi 4096 couleuren at your service! We zetten nu een DigiPaint-achtig kleuren-selectieblok op het scherm. U weet wel, zo'n rechthoek waarin de kleuren van de ene component op de x-as staan afgebeeld en op de y-as zijn de kleuren van de andere component afgebeeld. In totaal dus 15*15=256 kleuren op uw beeldscherm!
Hierna wordt een teller gestart welke tot een miljoen telt. Als dit voorbij is dan worden het window en het scherm gesloten. Alle gereserveerde ruimte voor de custom bitmaps worden weer vrijgegeven met behulp van de functie FreeRaster(). Tot slot wordt alle libraries, IntuitionBase en GfxBase weer gesloten. Gaat u nu niet zitten denken, waarvoor dient de library GfxBase nu weer, dit zal in een van volgende afleveringen in detail worden uitgelegd. We hebben extra commentaar bij de listing ingesloten zodat het niet al te moeilijk moet zijn om het voorbeeld te begrijpen. Snapt u er desondanks niets van, don't worry, be happy (wat een cliché) we komen er nog wel aan toe in een volgende aflevering.

De volgende keer

Dan scheidt ons nog maar één ding van de volgende bladzijde en dat is... de literatuurlijst. Experimenteer en lees voorzichtig, do it right, do it fine, and take your time. See you around!

Johan & Johan

Het element ViewModes

De ViewModes bepalen de grafische mogelijkheden van het scherm. Een tekenprogramma zal geheel andere eisen stellen aan het scherm dan een tekstverwerker. Het is dus zaak om van tevoren te beslissen wat voor mogelijkheden men nodig heeft, dwz. veel kleuren, hoge resolutie, grote flexibiliteit, etcetera.
In het volgende verhaal zullen we veelvuldig de term ViewPort hanteren. Dit zal voor de meesten een nog onbekende term zijn. Aangezien we later in de cursus de ViewPort nog uitgebreid genoeg zullen behandelen volstaan we nu met de opmerking dat de ViewPort hetzelfde is als het scherm (Screen).
We onderscheiden negen verschillende modi welke ruwweg in 5 categorieën in te delen zijn.

Resolutie bepalende modi

- HIRES: zorgt voor een verdubbeling van de x-resolutie (640 pixels).
- LACE: zorgt voor een verdubbeling van de y-resolutie (512 pixels).

Kleurenaantal bepalende modi

- HAM: afkorting voor Hold And Modify. In totaal kunt u dan over 4096 kleuren op het scherm beschikken.
- EXTRA_HALFBRITE: bitplane nummer 6 wordt nu ook gebruikt om een scherm met 64 kleuren te krijgen.

Dual playfield modi

- DUALPF: dual-playfield modus geselecteerd. Hierin kunt u twee schermen binnen één viewport maken. De achtergrondkleur is transparant zodat data in de tweede playfield zichtbaar worden.
- PFBA: een bitsetting welke de prioriteit in de dualplayfield modus bepaalt.

GenLock modi

- GENLOCK_AUDIO &
- GENLOCK_VIDEO: de achtergrondkleur wordt vervangen door het toegevoegde videobeeld. Hiervoor heeft men nog additionele hardware nodig (de zgn. genlock).

Overige modi

- SPRITES: in het scherm mogen sprites gebruikt worden.
- VP_HIDE: 'viewport hide'. De viewport wordt niet weergegeven. Deze flag is gezet als de viewport door een ander is bedekt.

Literatuurlijst

Berry, J.T. (1986), Inside the Amiga with C, Howard W.Sams & Comp., Indiana (USA) ISBN 0-672-22468-2
Bleek, Jennrich & Schultz (1989), Amiga Intern 2.0, Bruna uitg., Utrecht ISBN 90-229-3683-X
Bleek, Jennrich & Schultz (1988), Het grote Amiga C boek, Bruna uitg., Utrecht ISBN 90-229-3483-7
Dittrich, Gelfand & Schemmel (1987), Amiga Intern, Bruna uitg., Utrecht ISBN 90-229-3473-X
Jennrich, Trapp & Weltner (1987), Het supergrafiekboek voor de Amiga, Bruna uitg., Utrecht ISBN 90-229-3470-5

AmigaDOS en batchbestanden

Wilt u ook zo graag leuke features toevoegen aan uw gewone DOS, zoals routines die voor net geformatteerde printeruitvoer zorgen of routines die voor een alternatief DIR-commando zorgen? En u bent niet echt handig in programmeertalen als C of MODULA2? Wilt u meer met uw DOS doen dan u nu kunt? Let dan nu goed op want onderstaand verhaal is voor u geschreven.

Dit artikel is een toevoeging aan onze beëindigde Amiga DOS-cursus. We hebben toen de werkelijke kracht van één DOS-commando niet duidelijk genoeg naar voren gebracht. **Execute** was (en is) tot veel meer in staat dan we deden vermoeden. Even terug naar het verhaal omtrent Execute. Het is mogelijk om met dit commando een programma, of liever een file waarin DOS-commando's zijn opgenomen, uit te laten voeren. Zo'n file noemt men een 'batchbestand'. Een simpel voorbeeld:

```
IF EXISTS df2:
  PATH df2: ADD
ENDIF
```

Dit voorbeeld is erg simpel maar illustreert wel waartoe batchbestanden in staat zijn. Het bovenstaande verhaal controleert of volume 'df2:' aanwezig is, wat in ons geval de tweede drive zou zijn, en neemt deze op in de path lijst als dit het geval is. Nu zult u zeggen: 'Tja, redelijk simpel, maar waarvoor gebruik ik dat nou?' Terecht en onterecht. Terecht voor zover het de bruikbaarheid betreft. Eén keer gebruiken, bijvoorbeeld in de startup-sequence, is genoeg. Toch zijn er legio andere voorbeelden te bedenken waarvoor u batchbestanden kunt gebruiken. Gebruikt u vaak een programma waarvoor u maar enkele commando's in dient te typen? Maak een batchbestand die deze acties uitvoert en vervolgens het programma opstart. Bekijk onderstaand voorbeeld eens en oordeel daarna.

```
ASSIGN FONTS: df2:fonts
STACK 10240
PAGESETTER
```

Dit voorbeeld is puur illustratief dus typt u niet over. Als u 'pagesetter' op wilde starten zou u drie commando's in moeten typen. Eerst 'assign'. Daarna 'stack' en vervolgens 'pagesetter'. Gebruik het voorbeeld en typ dan in 'execute page', vooropgesteld dat de commando's in het batchbestand 'page' zouden zijn opgeslagen. Het is ook mogelijk om vanuit de SHELL gewoon 'page' in te typen. Zet het script-bit met behulp van het commando 'protect' en typ alleen 'page' in.

Nog meer met batchbestanden

Als het bovenstaande alles zou zijn, dan waren batchbestanden minder bruikbaar. Er zijn echter nog een aantal extra 'features' welke batchbestanden bijzonder bruikbaar maken.
Het is mogelijk variabelen binnen een batchbestand te gebruiken. Hiervoor zijn een aantal instructies of opdrachten beschikbaar welke Execute vertellen hoe de variabele heet en wat ze er mee moet doen. We kunnen zes instructies onderscheiden, te weten,**.key, .def, .dol, .bra, .ket en .dot**.
Het is u waarschijnlijk wel opgevallen, elke opdracht dient te beginnen met een punt ('.'). Dit is echter niet een standaard regel daar u dit zelf in kunt stellen. Hierop komen we straks terug.Hoe gebruiken we nu dergelijke variabelen in een batchbestand?

Declaratie van variabelen

Execute dient ten eerste te weten welke variabelen u wilt gebruiken. Hiervoor dient de '.key'-instructie. Deze dient als eerste te worden opgegeven in een batchbestand. Vermeld de '.key'-opdracht, en geef daarna, gescheiden door een spatie, de variabelenaam op. Wilt u meerdere variabelen gebruiken, dan zult u de variabelenamen, gescheiden door komma's, achter elkaar op moeten geven. Hoe ziet een dergelijk batchbestand er uit?

Bestand 002, DEMO1.BATCH

```
.key filename,targetfile
type <filename> to
    <targetfile> number
```

Dit voorbeeld activeert u door in te typen,

```
'execute demo.batch
     [naam_van_de_file]
     [doel_file_naam]'
```

of als u gebruik maakt van de script faciliteit van de SHELL,

```
'demo.batch
     [naam_van_de_file]
     [doel_file_naam]'
```

Zoals u al gezien heeft, dient u de variabelen van de tekens '<' en '>' te voorzien. Ook deze tekens staan niet vast, hierop komen we later nog terug.
Deze tekens en het tussenstaande worden bij de uitvoering van het bestand vervangen door de opgegeven filenaam en doelfilenaam.

Defaultwaarden

Het kan gebeuren dat u een defaultwaarde wilt gebruiken. Als de gebruiker geen invoer geeft, zal het batchbestand automatisch een waarde aan de parameter hechten, namelijk de waarde die u als default opgegeven hebt. Hiervoor dient u de '.def'-instructie te gebruiken. Achter deze instructie geeft u de parameter op, welke u natuurlijk met '.key' geïnitialiseerd heeft, en vermeld daarachter, gescheiden door een spatie, de defaultwaarde. Algemeen gesproken heeft de instructie de volgende vorm,

```
.def [variabele]
    [default_waarde]
```

Om nu het geheel wat duidelijker te maken, nemen we het voorgaande voorbeeld. We nemen hierbij aan dat de uitvoer standaard naar de printer wordt gestuurd.

```
Bestand 003, DEMO2.BATCH
------------------------------------
.key filename,targetfile
.def targetfile prt:
type <filename> to
    <targetfile> number
------------------------------------
```

Voor dit soort default-opdrachten is er een simpelere methode. Gebruik het dollarteken ('$') als scheidingsteken tussen de variabele naam en de defaulttekst. Weer even terug naar het voorgaande voorbeeld. We passen deze een klein beet-je aan en zorgen ervoor dat de uitvoer default naar de file 'DEFAULT.-OUT' gaat.

Bestand 004, DEMO3.BATCH

```
------------------------------------
.key filename,targetfile
type <filename> to
    <targetfile$default.out>
    number
```

Het dollarteken ('$') is het scheidingsteken tussen de variabelenaam en de defaulttekst. Dit is niet persé noodzakelijk. Er bestaat een mogelijkheid om dit te vervangen door een ander teken.

Vervangen van de standaard tekens

Zoals u hierboven al gezien heeft, heeft Execute, het commando wat voor de executie van het batchbestand zorgt, het dollarteken nodig om te kunnen onderscheiden waar de defaulttekst begint. Het kan echter gebeuren dat u dit teken al in uw filenaam gebruikt heeft. Dan zal bovenstaand voorbeeld niet werken. Met dit probleem is rekening gehouden.
De instructie '.dol' zorgt ervoor dat het standaard dollarteken vervangen wordt door een door u zelf opgegeven teken. Bijvoorbeeld '.dol !'
Dit zorgt ervoor dat u nu het uitroepteken dient te gebruiken om de scheiding tussen variabelenaam en defaulttekst aan te geven.
Op soortgelijke wijze als met het dollarteken kunt u ook de punthaken '<' en '>' vervangen door alternatieve tekens. Het commando '.bra' dient om de linker punthaak '<' te vervangen, en '.ket' dient om de rechter punthaak '>' te vervangen.
Tot slot is ook de punt '.', die voor elke instructie dient te worden gezet, te vervangen. De instructie '.dot' zorgt hiervoor. Bijvoorbeeld '.dot @'. Wilt u hierna de hierboven besproken instructies nog gebruiken dan dient u ze aan te roepen door het '@'-teken ervoor te vermelden. Bijvoorbeeld '@key file'

Commentaar in batchbestanden

Ook in een batchbestand kunt u commentaar ter verduidelijking van het geheel opnemen. Hiervoor gebruikt u de punt ('.'). Zorg er wel voor dat er een spatie tussen de punt en uw tekst staat. Heeft u de punt met behulp van het '.dot'-commando vervangen door een ander teken, dan zult u natuurlijk het teken moeten gebruiken dat u achter '.dot' vermeld heeft, in plaats van het puntteken.

Om het geheel te verduidelijken hebben we een batchbestand geschreven waarmee u een listing net geformatteerd naar de printer kunt sturen. Bestudeer het en typ het in. Voer het bestand uit met het commando 'Execute' of gebruik de scriptfaciliteit van de SHELL.

Batchfile demo. PrintFile Creation Utilty

```
.key command
.ket (
.bra )

echo >par: "
    PFCU v1.00"
echo "Copying file to
    parallel.device"
echo >par: "Filename
    :" NOLINE
list >par: (command) KEYS
echo >par: "Reference date
    :" NOLINE
date >par:
echo >par:
    "-------------------------
    --------------------------
    --------------------------
    ----------"
type >par: (command) number
echo >par:
    "-------------------------
    --------------------------
    --------------------------
    ----------"
echo >par: "End of file "
    NOLINE
list >par: (command) QUICK
```

Johan & Johan

BulletinBoards

We hebben als redactie van vele kanten gemerkt, dat er steeds meer behoefte is aan telefoonnummers van bulletinboards. Er blijken meer en meer gebruikers te komen, die graag gebruik maken van dit medium. De lol van een modem aan de computer is er dan al gauw af, als je geen nummers hebt. De redactie is eens een paar avonden aan het 'modemen' gegaan en heeft een aantal nummers van bulletinboard geselecteerd, die er interessant uitzagen voor het gebruik met een Amiga.

Omdat de bazen, sysops genoemd, nogal eens besluiten om andere telefoonnummers te nemen, of zelfs ineens te stoppen, kan de redactie niet met zekerheid zeggen of de BBS nog wel bereikbaar is.

Werkwijze

Hoe gaat het in z'n werk? Om te beginnen heb je dus een modem nodig. In het overzicht staat op welke baudrate (snelheid) een BBS werkt, en waarop je dus je modem op moet aanpassen. En verder hebben we de naam en het telefoonnummer vermeld. Hoe je een BBS benaderd, hangt helemaal van je programma af. Maar het doorlopen ervan is bijna bij ieder board hetzelfde.

Modemkeuze

En wat als je nog geen modem hebt? Er zijn een aantal mogelijkheden bij de aanschaf van een modem. Het ligt allereerst aan het type computer. Voor de Commodore 64 en de 128 ligt het allemaal wat ingewikkeld. Daarvoor zijn eigenlijk alleen de modems van Tron geschikt.
Dit zijn goede modems met bijgeleverde software.

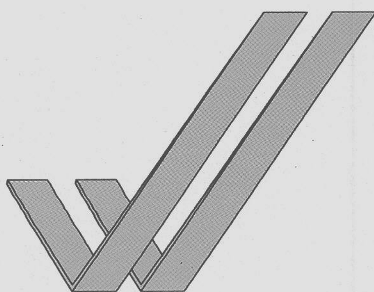
Voor de Amiga ligt het natuurlijk aan het type welk modem je aanschaft. De 500 is niet geschikt voor inbouw van een dergelijke apparaat, wat dus altijd betekent dat je het apparaat met een draadje (serieel) aan je Amiga moet hangen. Daarvoor zijn verschillende typen leverbaar. Meestal worden ze dan zonder software geleverd. Voor een geschikt programma kun je in de PD-series vrij goedkope programma's vinden, die zeker zo goed zijn als de meer professionele.
De Amiga 2000 heeft meer mogelijkheden. Ook hier kan een extern modem worden aangesloten. Maar bij het gebruik van een bridgeboard is het natuurlijk ook mogelijk een insteekmodem aan te schaffen. Voor de Amiga kant zijn er minder modellen als kaart leverbaar. We zagen al een model van Supra, dat perfect werkte.

Nog even een laatste opmerking over de BBS-lijst:
De redactie weet het goed dat we hier een klein gedeelte publiceren. Voor een compleet overzicht hadden we waarschijnlijk een halve Commodore Info nodig. Veel plezier met het modemen!

BBS-lijst

23	ACCESS BBS	050-128536
23	ACCOM-BBS	072-154426
23	AINEX-RBBS	033-633916
23	Amiga BBS Extreme	072-401889
123	Amiga Board	020-900800
23	Amiga Highlight	072-400971
123	Amiga Systems	020-6643379
23	BBS MEGA-NET	020-476992
123	BBS Opus SCUM	01185-2213
123	BBS Pendrecht	010-4102033
1234	BBS Rijswijk	070-3962340
23	Def Board	04998-97759
23	Develsteyn's Castle	078-196010
123	DLS-BOX Amersfoort	033-722161
124	Drechtstad HanWi	078-213314
123	Dutchman BBS	020-316873
1234	ENSING BBS	05904-1913
123	Filosoft/programmers	050-412288
23	Financial Information	03432-3233
14	Flevo BBS	03240-61409
123	Funny BBS	01620-37563
123	Futura BBS	075-352035
123	GAME BOARD	08851-15661
123	Hotline 1	020-891014
124	HUCO BBS	01804-30785
123	Hyper Castle BBS	078-164626
3	Magic Systems	01718-26936
123	McBAUD	010-4603947
123	MeerMail BBS	02503-41441
12	Milieu Bulletin Board	05179-1888
23	Mind BBS	05769-415
123	Ned. Net Nieuws	03480-24801
123	NEDTP / BSAH	070-3904955
23	NeijCo & Co	02040-126938
14	NIX BBS	020-197769
1234	NNN Positivo Board	05787-2223
1234	Opus Groningen	050-181631
1234	Opus GTH	035-61643
23	PC-Prof	01736-6981
23	PC-Square	079-424107
123	Perfect Board Int	020-962382
123	Phantasy-Board	070-3253586
123	PHCC-Enschede	053-301064
23	Pioneer BBS	070-3293952
123	QuickBBS Friesland	05130-25641
123	RemoteAccess	05109-4458
1234	Reset UNITED	02510-24874
23	SCAN-BBS	071-416627
1234	SYNCNET BBS	035-237178
123	Skyline BBS	010-4558300
1234	Soft Airmail	053-339781
123	The Final Countdown	085-231945
23	The Future BBS	020-279245
23	The GIT-Network	013-560209
123	The Higher BBS	023-321129
123	The Lair	079-312293
123	The Lamp	01717-3263
23	The Mailbox	040-122083
123	The MCC Board	070-3872485
123	WestPoint Castle	010-438254812

Legenda:
(1) V21 (300)
(2) V22 (1200)
(3) V22bis (2400)
(4) V23 (1200/75)



PD-hoek

In het vorige nummer van Commodore Info berichtten we al van een aantal goede programma's in de Public Domain-series. Daarin legden we ook uit wat PD-software nu eigenlijk is. We willen met die categorie doorgaan in de vorm van een eigen rubriek. Daarmee gaan we niet in de voetsporen van andere bladen voort met alleen Fred Fish en zijn grote collega's, maar willen we ook graag de wat kleinere "jongens" aan het woord te laten.

In deze rubriek bespreken we kort een aantal interessante programma's en vermelden op welke disks deze te vinden zijn. Wanneer het de moeite waard is, bespreken we een programma wat ruimer, zoals we de vorige keer met bijvoorbeeld PowerPacker deden.
Deze keer is de Nederlandse UGA van Ron Fonteine aan de beurt. Hij brengt verschillende series op de markt: Utilities, Newsflash, verschillende Shareware en binnenkort ook Power Utilities. Alle software die Ron Fonteine op zijn UGA-diskettes publiceert is PD of shareware.
Voor deze aflevering bekeken wij de eerste drie utility-disks en een Newsflash.

Utilities 1:

Twee programma's die direct opvallen in het menu zijn **Bootem** en **Bootem Dirty**. Beide programma's zijn bedoeld om een programma direct op te laten starten. Daarvoor kan worden ingegeven welk programma dat moet zijn, waarna die gegevens vervolgens in de bootsector worden gezet. Na het inleggen van de diskette staat dan binnen enkele seconden het programma in het geheugen. Als het programma beëindigd wordt, wordt de startup-sequence alsnog geladen.
Cli-Wizard is een handig stukje shareware dat vergelijkbaar is met Cli-mate, Disk-master en meer van dit soort programma's. Cli-Wizard heeft heel veel functies beschikbaar direct boven de statusregel. Heel handig is dat de Cli-Wizard ook als een menu kan worden gebruikt van waaruit programma's zijn op te starten. Daarvoor worden dan de namen ingegeven die in dezelfde hokjes verschijnen als Dir en Read, maar dan onder de statusregel. Door het aanklikken van een naam wordt het programma direct gestart. Dat kan ook van een tweede (externe) drive zijn. Het enige nadeel van deze Cli-Wizard is dat het de inhoud van maximaal een diskette kan laten zien, in tegenstelling tot de andere genoemde programma's.
Show-fonts is een handig programma om vanuit de CLI direct de fonts op het scherm te laten zien. Het programma werkt gewoon in een window en kan een font-directory laden. De fonts kunnen worden doorlopen met een menu dat onderin het venster verschijnt. waarbij de lettertypen cursief, vet, ondersteept worden voorgesteld.
Verder staan op deze schijf een aantal crunchers, programma's voor de bootsector, voor het draaien van muziek op de achtergrond, het omzetten van IFF-muziek naar 'raw data' en nog veel meer.



Boot'em

Utilities 2:

Op iedere diskette staan ze tegenwoordig: de **viruskillers**. Soms zie je door de bomen het bos niet meer. Op deze diskette staan er ook een paar met allemaal hetzelfde doel: virussen opsporen en verwijderen. Er staat ook VirusX op weliswaar in versie 2.01. En zoals dat op veel utility-diskettes de gewoonte is, hierop ook een **slideshow**. Deze maakt alleen geen gebruik van de afgezaagde mogelijkheden. Zo is een van de opties het in blokjes laten opkomen van een IFF-plaatje over een ander heen. Een effect wat we wel eens bij televisie-programma's zien.
Popcli is inmiddels heel bekend doordat hij ook op andere series voorkomt. Het is en blijft een handige utility. Eenmaal geïnstalleerd kan een nieuw CLI-venster met (Amiga-links)-Esc worden geopend. Aangezien het programmaatje resident in het geheugen zit, kan het met ieder ander programma worden geladen en op ieder gewenst moment worden geactiveerd.
Een **monitor** en een aantal **view-files** (onder andere voor tekstfiles en geheugen)

maken eveneens deel uit van deze diskette.

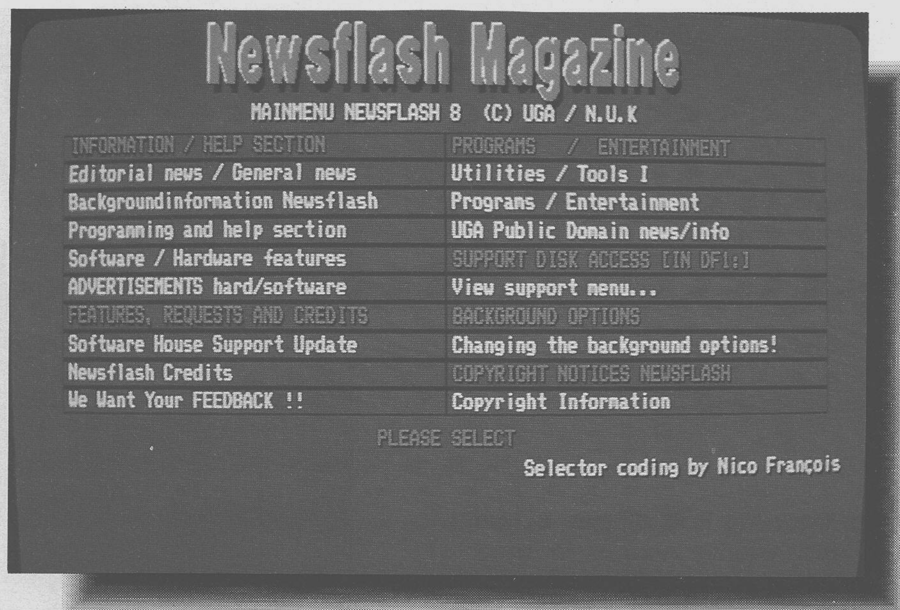
Utilities 3:

Op deze diskette komen we opnieuw een paar anti-virusprogramma's tegen. **Virusprotector IV** mag dan wel niet zoveel virussen herkennen, het is prachtig uitgevoerd en werkt perfect. Een aardigheidje dat UGA direct al bij het opstarten laat zien, is een muis die spontaan veranderd in een vliegtuig dat om zijn as draait.
Ook op deze diskette **Showfont**. Maar deze versie laat een hele tekenset tegelijkertijd op het scherm zien, dus letters, cijfers en speciale tekens.
Voor de muziekliefhebbers biedt deze diskette **ST-Player**. Met dit programma kunnen geluiden worden geladen en gespeeld vanuit een handig venster.
Hermit is een resident programma dat het huidige beeld kan wegschrijven naar ram: of een diskette. Bij gekleurde IFF-plaatjes is er ook de cycle-optie mogelijk. Het geheel wordt met behulp van een paar hotkeys bestuurd.
Disksalv kan files herstellen. Het werkt net als Diskdoctor vanuit de CLI, maar is krachtiger.

Newsflash

De **Newsflashes** van UGA zijn van heel ander kaliber dan de utility-schijven. Dat moet ook wel, omdat Newsflash in feite een tijdschrift op diskette is. We noemen de serie hier toch, vanwege de vele programma's die Ron Fonteine er óók opzet. Hij gaat zelfs zover, dat hij, vanwege de vele goede programma's, er nog een diskette bij maakt. Dat wil dus eigenlijk zeggen dat een aantal afleveringen van Newsflash uit twee diskettes bestaat. Fonteine noemt de tweede diskette **'Support disk'**. We gaan hier verder niet in op de inhoud, maar willen wel graag kwijt, dat deze diskettes beslist aanraders zijn voor wie helemaal bij wil zijn. Ze barsten van de foefjes en hulpjes, die ook gebruikt kunnen worden voor eigen diskettes, omdat het hier ook om PD en shareware gaat. Met één diskette kregen we na een reset (Ctrl-Amiga-Amiga) de hand met de diskette niet meer op het scherm, maar een melding over Newsflash... (een voorbeeldje)

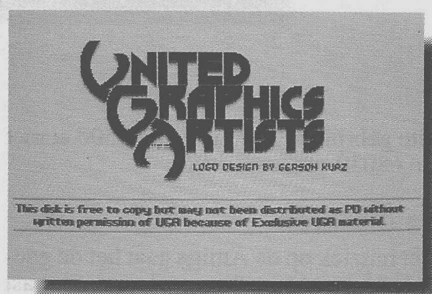
Ron Fonteine gunde ons een kijkje in de toekomst met het shareware-programma **'Sliding'**. Het woord sliding doet denken aan een diashow met IFF-plaatjes, maar dat is het beslist niet. Terwijl op de achtergrond een muziekje te horen is, zien we op het scherm een plaatje verschijnen. Het volgende moment wordt het plaatje in zo'n veertig stukjes verdeeld. En zie er nu maar eens weer het complete plaatje uit te toveren. Dat valt lang niet mee.



Newsflash

Het begin is ook hier het moeilijkst, omdat er een hokje knippert, waarin perse het juiste stukje plaatje in moet worden geplaatst. Het hokje dat dan vrij komt is daarna weer aan de beurt, enzovoort. Is dit plaatje goed gelukt, dan gaan we naar de tweede. Een leuk spel dat enig doorzettingsvermogen van de speler eist.

Ron Fonteine is 's-morgens bereikbaar onder telefoonnummer: 03404-52987.



Nieuwe versie VirusX

In vorige nummer schreven wij over PD-viruskillers, waaronder VirusX 3.2. Daar maakten we ook al melding van de nieuwe versie: 4.0. In dit nummer een korte uiteenzetting van de mogelijkheden van VirusX 4.0 (Fish 287) van programmeur Steve Tibbet.
Verstandig is om VirusX 4.0 op te nemen in de startup-sequence, maar dat geldt natuurlijk ook voor de andere viruskiller en -finders.
Nieuw bij versie 4.0 zijn de diverse opties die bij het starten kunnen worden meegegeven. We sommen ze hier even op:
'-c' en '-k' zijn bedoeld voor het controleren van de CoolCapture en de KickTagPtr. Bij de oudere versies werden deze controles voor die van de drives uitgevoerd.
'-a' laat de VirusX-balk bovenin beeld zien. (Standaard bij de oudere versies)
'-q' heeft overeenkomsten met het woord 'quit'. Als VirusX 4.0 met deze optie wordt gestart, check hij de drives en verdwijnt daarna uit het geheugen.
'-r' is alleen van belang voor gebruikers van een Amiga met 1 MB chip-RAM.
Twee opties die bij 4.0 als nieuw worden aangegeven, maar het eigenlijk niet zijn, zijn '-x' en '-y'. Acher de x en de y komen de coördinaten te staan van de WorkBench. VirusX -a -x320 -y100 houdt in dat het programma op de coördinaten 320,100 op het scherm komt te staan. Wanneer er bijvoorbeeld al een klokje bovenin de WorkBench staat, is het handig om VirusX helemaal rechts onderin neer te zetten. Na een beetje uitvogelen moet dat lukken.
VirusX is nu ook geschikt voor file-virussen die zich aan een programma hangen. Hij kan de meeste van deze virussen herkennen en verwijderen. Kleine problemen onstaan bij het Xeno-virus. VirusX herkent het, maar kan het niet direct van de diskette halen. Dat moet gebeuren met een extra programmaatje met de naam 'KillVirus', wat ook in dezelfde directory staat als VirusX.
Aangezien het aantal virussen nog steeds toeneemt, zal het wel niet lang duren voordat er weer een nieuwe versie verschijnt.

W.S.

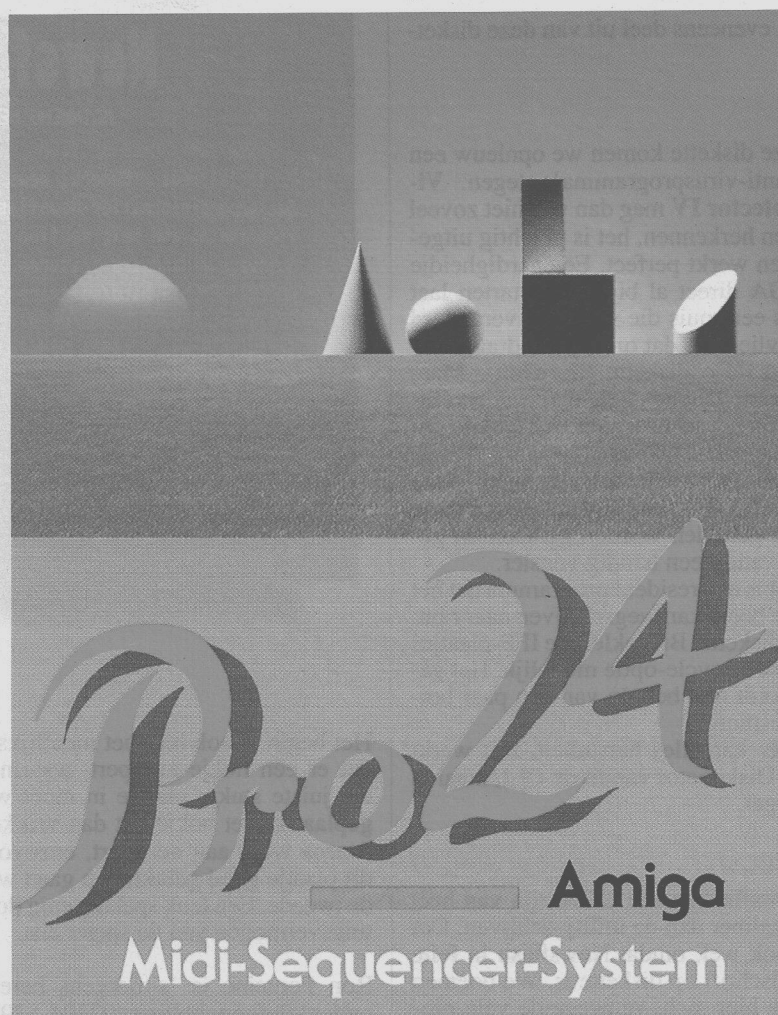
Amiga op de CeBIT

Commodore had dit jaar op de CeBIT weinig nieuws te melden. Dat bleek ook op de persconferentie: verhalen over de 64 als de meestverkochte computer kenden we natuurlijk al. Wel nieuw (in Europa) was een Amiga 2500 met een 68030-processor. Wensen van een nieuwe Kickstart 1.4 en een Amiga 3000 gingen nog niet in vervulling.

Commodore neemt een zeer sterke positie in op de Duitse markt. De verwachte Amiga met CD-ROM was er niet en de 'nieuwtjes' die er waren, waren bij het publiek al min of meer bekend. Zo was er de PC 60-III met een 80386 CPU en de Amiga 2500 met de Motorola 68030. Dat deze laatste machine er in Hannover bij zou zijn, kwam eigenlijk niet als een verrassing.

Op de CeBIT werd nog eens duidelijk dat Commodore met de Amiga de professionele kant op wil. De gepresenteerde 2500 met 68030-processor is tweemaal zo snel als zijn oudere broertje met een 68020. Dat ligt niet puur aan de processor alleen, maar ook aan de toepassing van een 32-bit databus en het Memory Management. Deze 2500 werd gepresenteerd met een rustig beeld van een Eizo 9060-S beeldscherm. Dit topmodel in de Amiga-serie zal in Duitsland ongeveer 8990 DM gaan kosten.

CBM ziet dit systeem als de basis voor professionele toepassingen als DTP, Desktop Video en presentaties. Het heeft 3 MB geheugen ter beschikking en een harddisk van 40 MB met een SCSI-controller. Een 3,5 inch diskdrive is standaard, terwijl er nog plaats is voor een extra optionele 3,5- of 5,25 inch drive. Verder zijn er vier slots voor Amiga-kaarten, vier AT-slots, een Video-aansluiting voor een PAL-videokaart of genlock. De andere aansluitingen zijn standaard-Amiga. In feite gebruiken sommigen de 2500 al met een 68030-processorkaart.



Steinberg Pro 24

De anderen

Met muziek op de Amiga gaat het de goede kant op. Al eerder werden we verrast door pakketten als MusicX, maar de prijs is vaak behoorlijk hoog.

Steinberg presenteerde op de CeBIT de Amiga 24 sequencer, die veel overeenkomsten vertoont met Pro 24 voor de Atari ST. Het gaat hier om een professionele 24 sporen midi recording-programma. Tot de mogelijkheden behoren onder andere 96 ppq resolutie, grid-editor, drum-editor en een MIDI-mixer.

Amiga Pro 24 loopt op iedere Amiga (Kickstart 1.2 of hoger) met minimaal 512 KB, hoewel 1 MB wordt aanbevolen. In Nederland gaat de versie *f* 595,- kosten zonder MIDI-interface en wordt geleverd door James B. International (010-4588166).

In de vorige Commodore Info meldden we dat **Gigatron** naar alle waarschijnlijkheid een portable Amiga zou presenteren. Dat bleek helaas niet het geval. Wel had de firma een printbuffer in zakformaat.

Veel aandacht was er voor videotoepassingen. Verschillende bedrijven showden genlocks en digitizers. Voor het eerst werd er een complete oplossing voor de muziekvrienden getoond: een Amiga 500 met MIDI-interface, een 16-Bit keyboard van **Kawai** en de nieuwe Steinberg software: Pro 24. De firma Kawai, die dit systeem verkoopt (in Duitsland) wil de komende tijd met een serie software komen, die is afgestemd op het Kawai-muzieksysteem.

Muis

Een ander intessante vinding was een kabelloze muis voor de Amiga. De muis is voorzien van een batterij, waarmee 10 uur achtereen kan worden gewerkt. De normale draad is vervangen door een infrarode zender en ontvanger. De reikwijdte bedraagt ongeveer 1,5 meter. Om de batterij een beetje te sparen is er een tijdmecha-

nisme ingebouwd, dat er voor zorgt dat de muis uitgaat na vijf seconden niet gebruikt te zijn. De prijs van het staartloze beestje lag op de CeBIT rond de 350 gulden.

Gold Disk

Veel nieuws was er bij **Gold Disk**, de maker van onder andere Pagesetter en Professional Page. Gold Disk had tot de CeBIT gewacht met de introductie van de nieuwe versies van deze twee programma's.
Professional Page is op vele punten verbeterd. Het programma is nu ge-update van versie 1.3 naar 2.0 en kan nu de inhoud van boxen roteren, twee pagina's tegelijk op het scherm laten zien en het ondersteund style-sheets, zoals we dat eigenlijk alleen maar bij professionele DTP-pakketten voor de Apple en de PC tegenkomen. Verder kan de nieuwe versie PostScipt-files inlezen en die op het scherm laten zien. De prijs van de nieuwe versie komt op ongeveer 1800 gulden.
En zoals het zo langzamerhand de gewoonte is geworden, heeft Gold Disk ook Professional Draw ge-update naar dezelfde versie als Prof Page. Niet veel verschillen met versie 1.3, maar toch werd het programma sneller en liet men een aantal nieuwe mogelijkheden voor de tekst zien.
Voor de gebruiker die minder geld heeft te besteden, heeft Gold Disk Pagesetter II uitgebracht. Deze versie gaat nu ook de professionele kant op en doet qua features sterk denken aan Pagestream, het DTP-pakket van concurrent Soft Logic.



Netwerk

Dat de Amiga 500 niet wordt vergeten, bewijst **Advanced Computer Design** (ACD). Dit bedrijf was op de CeBIT te zien met een netwerk waarin een aantal Amiga's waren opgenomen. Daarbij speelden de 500 en de 2000 de hoofdrol, De kaart is geschikt voor aansluiting aan de expansiepoort. Binnenkort komt deze kaart naar Nederland, waar 3Gitaal (020-970035) uit Amsterdam het distributeurschap op zich zal nemen.

Hoewel de CeBIT van oorsprong het toneel van nieuwtjes was, bleek het dit jaar toch een beetje tegen te vallen, tenminste voor Commodore.
Maar niet getreurt: Van 10 tot 12 mei wordt de **Amiga 90-manifestatie** in Basel (Zwitserland) gehouden. Belangstellenden kunnen schrijven naar: Ami Shows Europe GmbH, Zugspizstrasse 2a, D-8011 Vaterstetten.



Over muizen gesproken

Iedere Amiga wordt standaard met een muis geleverd. Niemand vindt het gek dat er een muis bij zit en ook niet dat de Amiga-muis nu juist dit model heeft. Veel gebruikers gebruiken daarom te goeder trouw de meegeleverde muis, vaak zonder te beseffen dat er nog andere modellen voor de Amiga leverbaar zijn. Maar ook muizen verouderen en sommigen sterven een vroege dood.

Verschillen

In principe zijn alle reserve-muizen, zoals we ze maar even noemen, gelijk. De aansluiting, de toetsen en de lengte van het aansluitsnoer verschillen niet veel van elkaar. Alleen het model van de kast heeft een eigen karakter.

Foundation Hard-User International, een grote Amiga-club in het zuiden van het land, levert de Reis-muis van de Duitse fabrikant Reis-ware. Het model is iets korter dan dat van de standaard Amiga-muis. De Reis-muis is boller en past feitelijk precies in de muis van de hand. Het snoer is van iets mindere kwaliteit.

De redactie had de eer een tijdlang met deze muis te werken. Hij bleek goed te voldoen, reageerde erg precies en lag lekker in de hand: een goede vervanger voor de originele Amiga-muis. De prijs voor deze muis ligt rond de *f* 80,-.

Compleet

Voor nog geen honderd gulden levert Catronix een compleet pakket, als vervanging voor de Amiga-muis: de Contriver-muis. In de mooi uitgevoerde verpakking bevinden zich een muis-mat, een muishouder en de muis zelf. Het snoer is van betere kwaliteit dan die van Reis-ware. Het aantrekkelijke van deze vervanging is het complete pakket tegen een lage prijs.

Geniaal

Eurosystems uit Duitsland verkoopt de Genius-muis, voor velen bekend van de PC. Het model verschilt van de anderen door de rechte hoeken. Verder zijn de toetsen van zacht rubber, wat erg prettig is. De kwaliteit van deze kabel is ook vergelijkbaar met die van het standaard model. In Nederland wordt Eurosystems vertegenwoordigd door Hupra in Arnhem.

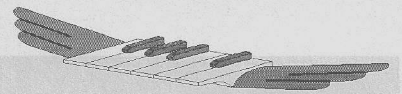
Conclusie

Voor iemand die zit te denken over de aanschaf van een andere muis is er genoeg keus. Let daarbij op een aantal zaken. Ligt de muis wel lekker in de hand, of is hij eigenlijk te kort? Hoe staat het met de kwaliteit van de kabel? En tenslotte de prijs. Wat krijg je voor hoeveel geld?

Het verdiend aanbeveling om eerst eens goed rond te kijken, voordat er een muis van de reserve-bank wordt gehaald. Of misschien toch weer een echte Amiga-muis? De keus is aan u.

W.S.

Inlichtingen:
Foundation Hard-User Int.: 040-417596
Catronix: 010-4507696
Hupra: 085-426716



MIDI in vogelvlucht

De digitale techniek en de computer hebben de laatste jaren een geweldige invloed gehad op de muziek. Er kunnen geluiden gemaakt worden die nooit eerder door iemand gehoord zijn. De composities hebben een wezenlijke verandering ondergaan, doordat er nu muziekpatronen gecomponeerd kunnen worden die te ingewikkeld zijn om door mensenhanden gespeeld te worden, maar die wel door de computers en synthesizers gespeelt kunnen worden.



Slechts enkele jaren geleden was de elektronische muziek het domein van de professional of de daarin gespecialiseerde hobbyist. Ondanks de uitgebreide mogelijkheden was er veel kennis van zaken nodig om een leuk muziekje aan de synthesizer te ontlokken. Anders kwam men niet verder dan een aap met een viool.
Nu heeft de elektronica een groot deel van de taken van ons overgenomen. Door de ongekende mogelijkheden van bijvoorbeeld sequencers kan zelfs de middelmatige muzikant nog fantastische symphonieën maken en ze zonder een noot te kunnen lezen ook nog op notenschrift zetten.
Naast de muziek zelf is er ook een enorme verandering gekomen in de muziekmarkt. Vele professionele muzikanten verdienen hun brood in de commerciële muziek. Niet alleen in de pop, maar met name muziek voor reclamespots en films. Eén persoon kan nu een geluid voortbrengen waar vroeger een heel orkest voor nodig was. Deze grote veranderingen zijn voornamelijk terug te voeren op de officiële invoering in 1983 van MIDI:de Musical Instrument Digital Interface.

Ingeburgerd

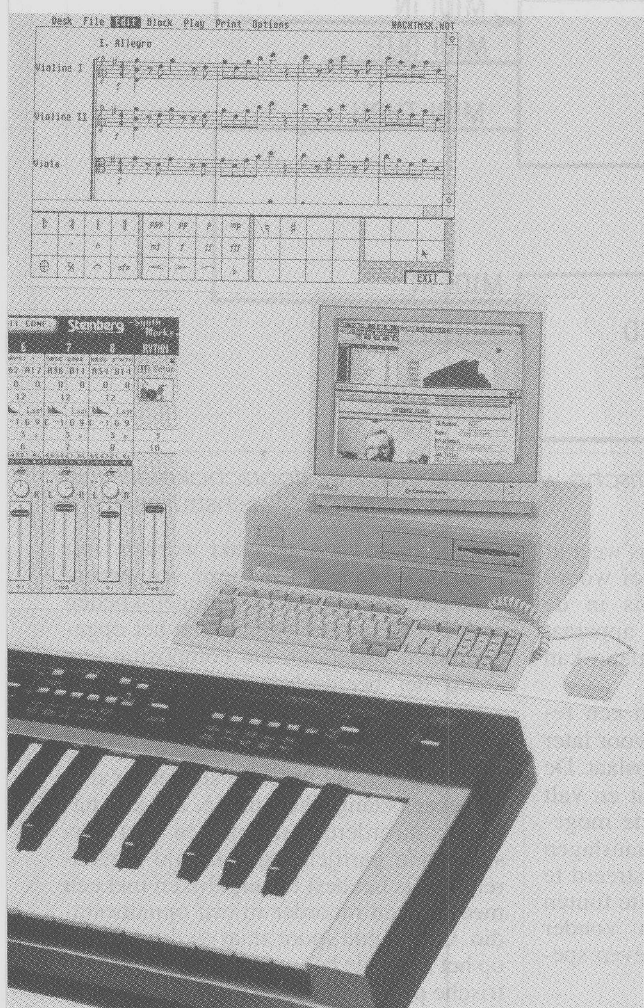
Iedereen heeft inmiddels de term MIDI wel eens gehoord en er wellicht wel eens mee gewerkt. Het is langzaam maar zeker een ingeburgerd begrip geworden onder muzikanten en computergebruikers. Wat het doet en hoe het werkt is echter niet voor iedereen duidelijk.
MIDI zal volgend jaar haar tienjarig bestaan kunnen vieren. Eind 1981 werd het eerste MIDI-instrument geïntroduceerd. Daarna is er een echte hausse van MIDI-instrumenten gevolgd. Deze geweldige groei was met name te danken aan snelle en goede afspraken tussen de verschillende muziekinstrumenten fabrikanten, die MIDI al in 1983 als frabieksstandaard erkenden.
Voor deze tijd waren er al veel mensen bezig met muziek en computers, maar iedereen ging zijn eigen weg. En aangezien er alleen nog analoge synthesizers waren, werd de ontwikkeling niet bespoedigd. Na de internationale erkenning van het MIDI-protocol werd de weg opengelegd voor een snelle introductie van digitale muziekinstrumenten.
In eerste instantie waren dat voornamelijk toetseninstrumenten en drumcomputers, maar de laatste jaren hebben we ook de eerste digitale gitaren en saxofonen mogen begroeten.

Communicatieprotocol

MIDI (Musical Instrument Digital Interface) is zoals de naam al aangeeft een interface. In de ruimste zin van het woord wil dat zeggen, dat het communicatie tussen twee of meer onderdelen mogelijk maakt. Om wat meer precies te zijn: MIDI is een standaard protocol van elektronische specificaties voor het communiceren van muziek. Het is dus niet meer en minder dan een (seriële) communicatietechniek. Net zoals bij vele computersystemen is MIDI zowel een hardware als een software standaard.

Die communicatietechniek kent bij MIDI drie belangrijke mogelijkheden.

- Allereerst Simultaan-besturing. Laten we eens uitgaan van een synthesizer, een drumcomputer en een sequencer (een opname- en afspeel- apparaat voor MIDI-informatie). Zonder MIDI zou het onmogelijk zijn deze drie instrumenten zinnig met elkaar te laten samenwerken. MIDI daarentegen kan deze instrumenten in perfecte harmonie met elkaar laten samenspelen. We noemen dit simultaan-besturing van andere MIDI-apparatuur
- Door het overzenden van MIDI-informatie kan de ene synthesizer met de ander meespelen. De overgezonden gegevens bevatten dan alle informatie die nodig is om het andere instrument te vertellen wat 'ie moet (mee)spelen. De elektrische piano kan 'praten' met de synthesizer of met de drumcomputer.
- De tweede mogelijkheid is opname en opslag. Door de perfecte registratie van de digitale codes die hiervoor wordt benodigd, is opnemen ook mogelijk. Dit brengt ons bij een van de belangrijkste eigenschappen van MIDI, namelijk dat de informatie over een stuk muziek opgeslagen kan worden in digitale vorm en dus naar een sequencer, een floppy of naar een harde schijf van een computer worden weggeschreven. De volgende dag bijvoorbeeld, kan de muziek opnieuw en in exact dezelfde vorm worden herhaald. Volledige composities kunnen als file weggezet en naar believen weer afgespeeld worden.
- En ten slotte editting. Alle met MIDI opgenomen muziek kan later met behulp van computerprogramma's aangepast worden.



Communicatie-poorten

De communicatie van de muziekinformatie die voor al deze zaken nodig is, vindt plaats via de drie mogelijke MIDI-aansluitingen: MIDI-in, MIDI-thru en MIDI-out. Zoals de woorden al aangeven komen bij de eerste de signalen binnen, bij de tweede gaan de signalen er alleen doorheen en worden dus weer uitgevoerd en dat brengt ons vanzelf bij de laatste communicatiepoort, die zorgt voor de uitvoer van MIDI-informatie.

De functie van de MIDI-in en -out zijn relatief makkelijk te begrijpen. Men kan hen het best vergelijken met de oren en de mond tijdens een communicatieproces. De ene zendt en de ander ontvangt. De functie van MIDI-thru vraagt daarentegen wat meer uitleg. In feite is deze uitgang grotendeels te vergelijken met de MIDI-out; het zendt informatie naar het volgende instrument. Het grote verschil is echter, dat deze communicatiepoort niet de door de eigen synthesizer opgewekte klanken kan versturen, maar slechts datgene kan herhalen wat door middel van de MIDI-in ontvangen is. Het praktische nut van deze uitgang wordt duidelijk, wanneer men drie synthesizers met elkaar wil verbinden. Met behulp van de MIDI-thru kan de informatie verzonden door midi-instrument 1, via midi-instrument 2 ook door instrument 3 worden ontvangen (zie illustratie 1). Hier komen ook de benamingen 'Master' en 'Slave' vandaan. De master bepaalt wat de slave gaat spelen. De Amiga met apart MIDI-klavier kan gemakkelijk de taak van master vervullen. Mits u maar over een compatibel MIDI-interface en bijbehorende software beschikt. Alle andere MIDI-instrumenten, zoals losse synthesizers, expanders en ritmeboxen worden dan tot slaaf gebombardeerd terwijl dirigent Amiga de stokjes stevig in handen houdt.

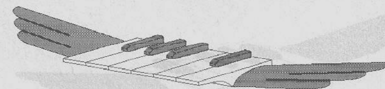
De MIDI-interfaces voor de Amiga zijn niet duur (ca. *f* 180,-). De 5-polige DIN-aansluiting is veel gemakkelijker te maken dan de lastige RS-232C (seriële) kabels. Toch moeten er wel enkele kwaliteitseisen aan de interface-elektronica gesteld worden anders gaat er te veel muzikale kwaliteit verloren.

Het aantal slave synthesizers kan echter niet onbeperkt worden uitgebreid. Vier of vijf slaven is meestal wel het maximum dat aan een master kan worden gehangen. Daarna treden er meestal veel communicatiefouten op zodat de informatie meestal ernstig gehavend arriveert op het eindadres.

Amiga interface

Een MIDI-interface (voor de Amiga) ziet er aan de buitenkant relatief eenvoudig uit. Een kastje met een seriële aansluitkabel voor de Amiga en de bussen MIDI-IN, MIDI-THRU en één of twee MIDI-OUTS.

Het verzenden van de informatie gebeurt door middel van zogenoemde MIDI-kabels. Ze zien er uit als de bekende DIN-kabels voor stereo-toepassingen (ze hebben ook een vijf-polige stekker). Verwissel ze echter niet anders kunnen er problemen ontstaan. Hoewel ze uiterlijk erg

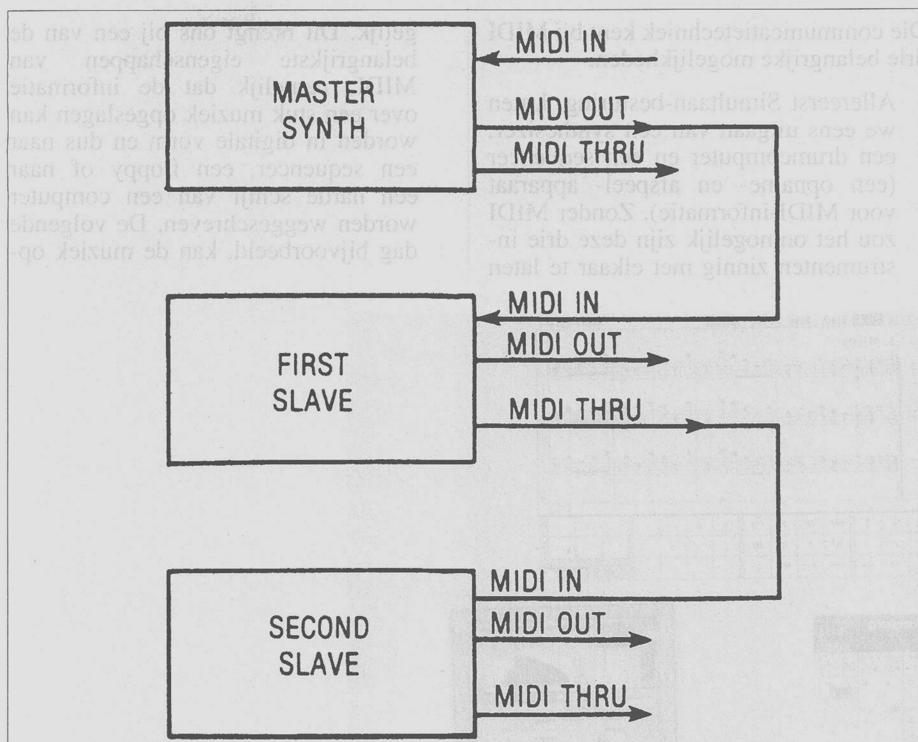


De Amiga-aansluiting

Voor het aansluiten van een Amiga 500, 1000 of 2000/2500 op een MIDI-netwerk zijn de volgende handelingen nodig:

- aansluiting van de seriële connector van de interface op de dienovereenkomstige Amiga-poort;
- de kabel vanuit de MIDI-OUT van het master-klavier gaat in de MIDI-IN van het Amiga-interface;
- via de MIDI-OUT-bussen van het computerinterface worden de MIDI-slaven aangesloten;
- de THRU-bus wordt voor het doorlussen of bij gebrek aan voldoende MIDI-OUT-bussen ingezet.

N.B.: Over de MIDI-THRU aangesloten apparaten (van MIDI-THRU naar MIDI-IN) spelen simultaan mee met het instrument dat de MIDI-output naar dit MIDI-apparaat zendt. Die via de MIDI-OUT naar MIDI-IN aangesloten instrumenten kunnen ook afzonderlijk aangestuurd worden.
Na het aansluiten laadt de gebruiker de besturingssoftware in en kan het componeren en het opnemen beginnen. Het grafisch interface op de Amiga laat precies zien wat er met de verschillende partituren gebeurt. Desgewenst kan er ook in notenschrift weergegeven of afgedrukt worden. Jammer is dat editors vaak met slechts een beperkt aantal synthesizers compatibel of sterk merkgebonden zijn. Let daarom terdege op welk MIDI- besturingspakket u nodig heeft.
Met MIDI betreedt de Commodore Amiga de arena van electronic composing en muziekuitvoering. Via relatief goedkope soft- en hardware-uitbreidingen kan de Amiga hele 'orkesten' zonder musici aansturen. Natuurlijk vereist dit alles wel enig talent van de (hobby-) componist, anders ontstaat er ondanks alle kracht niets anders dan een potporri van klanken.



Een schematische weergave van het doorschakelen van verschillende Midi-instrumenten.

veel op elkaar gelijken hebben ze een wezenlijk verschillend karakter.

Sequencers

We hebben zojuist de werking van de MIDI-communicatie wat verduidelijkt aan de hand van een voorbeeld van aaneen geschakelde synthesizers, maar zoals al genoemd is dit slechts één van de mogelijkheden van MIDI. Een wellicht veel belangrijkere functie van MIDI is de mogelijkheid de muziek informatie in digitale vorm op te slaan en vervolgens weer af te spelen. Dit heet met een mooi woord 'sequencing'. Een sequencer is in de ruimste zin van het woord een apparaat en/of software dat MIDI-informatie kan opnemen en afspelen.
Eigenlijk is het niets meer dan een recorder die alle keyboard-acties voor later gebruik als een MIDI-bestand opslaat. De kwaliteit van de sequencer staat en valt met diens nauwkeurigheid en de mogelijkheden van de **editor**. Alle aanslagen dienen exact herkend en geregistreerd te worden. De editor moet gemaakte fouten vlekkeloos kunnen herstellen zonder weer het gehele stuk over te hoeven spelen.

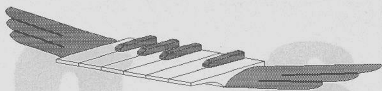
Grofweg zijn er twee typen te onderscheiden.

- Om te beginnen de hardware sequencer. Dit is een sequencer met een eigen behuizing, meestal met ingebakken software en al dan niet met een floppy disk drive.
- Ten tweede de software sequencer. Deze sequencer is geheel softwarematig en draait op een computer. Deze computer moet dan wel voorzien zijn van een MIDI-interface anders kan er niet met een MIDI-instrument worden gecommuniceerd. In de regel zijn deze softwarematige sequencers veel uitgebreider qua mogelijkheden en bieden zij zeer veel bedieningsgemak. Verder kunnen ze snel up-to-date gemaakt worden. Het grote voordeel van deze sequencers is dat zij meestal veel mogelijkheden bieden voor het editten van het opgenomen materiaal. De compositie kan op het beeldscherm worden aangepast.

Naast het opnemen en afspelen van MIDI-informatie heeft de sequencer nog een zeer belangrijke functie. Hij kan namelijk meerdere instrumenten met verschillende partijen tegelijkertijd aansturen. Het is het best te vergelijken met een meer-sporen recorder in een opnamestudio. Op het ene spoor staat de drumpartij, op het ander de bas, op het volgende elektrische piano etcetera. Ze spelen dus niet, zoals bij het voorbeeld van de master en slave allemaal dezelfde melodie, maar elke synthesizer speelt een andere partij. Op deze manier kan er dus een heel orkest worden weergegeven.
Men moet dan echter wel beschikken over meerder MIDI-instrumenten om deze partijen ook daadwerkelijk gelijktijdig te kunnen weergeven. Ten minste zo was dat tot voor kort.

Multitimbraal

Sinds enige jaren zijn er ook de zogenaamde multitimbrale synthesizers op de markt. Het verschil met hun voorgangers is dat deze instrumenten eigenlijk bestaan uit meerdere synthesizers in een apparaat. Deze instrumenten kunnen daarom tegelijkertijd een pianogeluid, een bas, een



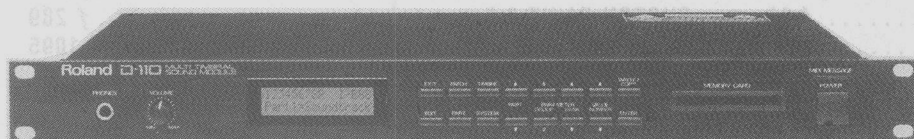


Tegenwoordig verschijnen er steeds meer synthesizers met ingebouwde sequencers met eigen floppy stations.

drum en dergelijke uitsturen. Het maximaal aantal sporen (tracks) bedraagt meestal 8 of 16. Deze synthesizers hebben soms al een eigen sequencer ingebouwd, maar komen meestal pas volledig tot hun recht met software-sequencer.
Multitimbraal is dus wezenlijk wat anders dan polyfoon. Twee termen die nogal eens door elkaar worden gehaald. Polyfoon wil zeggen dat men meerdere tonen (met de zelfde klankkleur) tegelijkertijd kan bespelen.

Software

In het bovenstaande hebben we de belangrijkste software voor MIDI al besproken: de sequence-software. Deze software is er in vele soorten en maten. Met name voor de Atari en de Apple Macintosh is er veel op de markt. De Amiga is de laatste jaren sterk in opkomst en de bekendste programma's, die voorheen alleen voor de Atari en de Mac beschikbaar waren zijn onlangs ook in Amiga-versies verschenen. De DOS-machines blijven voorlopig nog even achter. Naast het onderscheid in type computer varieert de sequence-software voornamelijk in de edit mogelijkheden en de manier van grafische representatie op het beeldscherm.
Editten wil in dit verband zeggen het aanpassen of maken van bestaande composities op het beeldscherm. Hierbij kan men denken aan het verplaatsen van stukken muziek in de tijd, de duur van noten veranderen, het volume aanpassen en nog veel meer. Voor al deze soms zeer complexe aanpassingen is een goede grafische weergave onontbeerlijk. Elke noot beschikt over een groot aantal variabelen, die ook bij een 24-sporen sequencer overzichtelijk moeten worden weergegeven. De grafische interface van de programma's kan behoorlijk verschillen en bepaalt grotendeels de prijs en het gebruikersgemak. Via iconen, notenbalken, timer-instrumenten en recordingwindows (record, sampling en play-back) valt alles keurig te besturen en te saven.
Naast de sequence software is er nog een aantal andere belangrijke groepen muziek-software.



Steeds vaker wordt het toetsenbord gescheiden van de synthesizer. Op deze foto ziet een multitimbrale muziek module zonder toetsenbord, zij kan uitsluitend via MIDI-signalen worden aangestuurd.

Amiga en MIDI

Een synthesizer is, wat oneerbiedig gezegd, eigenlijk niets anders dan een elektronische klankgenerator. Hetzelfde kan ook gezegd worden van de Amiga-soundchips, die eveneens aangeklede toongenerators zijn. Een goede synthesizer kan uiteraard meer dan onze Amiga. Het muziekinstrument beschikt over een echt klavier en kan een groot aantal op chips ingebakken (sampling) instrumenten zuiver ten gehore brengen. Verder zitten er allerlei interessante extra opties op zoals ritmeboxen, automatische accoorden, handen-geklap en het gelijktijdig laten horen van meerdere instrumenten.
Voordat men zo'n geavanceerde synthesizer in de muzikale vingers heeft gaan er heel wat lessen in zitten. En dan komt u dikwijls nog handen te kort. Met name bij dat 'handen te kort' kan de Amiga hulp bieden.

Muziek als elektronisch signaal

Zowel de Amiga 500, 1000 of 2000/2500 als de synthesizer wekken bewerkbare muzieksignalen op. Die signalen kunnen onderweg verder bewerkt (gemoduleerd) worden door software of hardware en zullen ergens naar toe gevoerd moeten worden. Dat 'naar toe' is bij voorkeur een goede stereo-installatie, want dan komt het gegenereerde geluid pas optimaal tot zijn recht.
De Amiga-soundchips en synthesizers genereren bepaalde grondtonen en geluidstrillingen die na bewerking een bepaald instrument imiteren. Hard- en software bieden de modulatie- en regelinstrumenten om de juiste klank(en) uit de opgewekte signalen te halen. In die signaalbesturing schuilt het addertje onder het muzikale gras. Je moet wel een volleerd technicus zijn om alle bewerkingen te kunnen doorgronden en ook daadwerkelijk te kunnen toepassen.
Zoals wel meestal het geval is, holt de gebruiksvriendelijkheid hijgend en puffend achter de snelle elektronische ontwikkelingen aan. Dank zij de vergaande digitale synthesizer- en interfacetechniek en de groei van het muziek-software-aanbod kan de Amiga nu veel lastige technische taken van de gebruiker overnemen. Draai uzelf echter geen rad voor de ogen: verstand van en gevoel voor muziek zult u altijd moeten hebben anders komen slechts de meegeleverde voorbeelden uit de dure hard- en software.

Muzieknotatieprogramma's

Eén van de grote voordelen van MIDI is dat de MIDI-informatie niet alleen vertaald kan worden in muziekklanken, maar ook in muziekschrift. Immers alle informatie over de lengte van de noot, de toonhoogte, de maat en het volume is voor handen. Muziektekstverwerkers bestaan al geruime tijd, de mogelijkheid tot het rechtstreeks invoeren en omzetten van MIDI-informatie naar notenschrift is echter nog een vrij jong terrein. Voor de PC en Apple zijn er echter al enige tijd goede en erg dure programma's op de markt. Atari en Amiga zijn tot nu toe helaas nog wat minder bedeeld, maar zoals het er nu naar uitziet zal hier snel verandering in komen.

Compositieprogramma's

Er zijn inmiddels programma's op de markt die componisten ondersteunen bij het maken van hun partituren. Het gaat zelfs zover, dat de computer gehele begeleidingspartijen kan genereren bij een vooraf ingebrachte melodie.

Patch-editors.

Een patch is niets anders dan de definitie van een synthesizergeluid. Hierin zijn alle waarden van de verschillende parameters (de variabelen die het geluid bepalen) gedefinieerd. Een patch editor biedt de muzikant de mogelijkheid om deze geluiden op overzichtelijke wijze aan te passen, om zo nieuwe ander geluiden te kunnen maken.

Bibliotheek-programma's

Het geheugen van de synthesizer is vaak te klein om alle geluiden die een muzikant wil bewaren op te kunnen slaan. Het gebruik van de harde schijf van een computer biedt dan ook vaak een welkome uitbreiding van dit geheugen. Voor een beetje enthousiaste hobbyist zijn 1000 geluiden geen uitzondering. Het steeds populairder wordende samplen draagt hier een behoorlijk steentje aan bij. Het is daarom uitermate handig om een goede orde aan te kunnen brengen in dit grote aantal geluiden. Dergelijk bibliotheekprogramma's zijn hiervoor speciaal ontwikkeld.
Zij bezitten over diverse mogelijkheden om geluiden te sorteren te selecteren etcetera.

Educatieve muziekprogramma's

Steeds meer wordt de computer (al dan niet in combinatie met een synthesizer) gebruikt voor muziek onderwijs. De computer vormt dan meestal een aanvulling op de lessen van de docent en stelt leerlingen in staat om thuis een flink portie zelfstudie te doen.
Als aparte groep kunnen we hier nog de programma's vermelden die niet gericht zijn op aansluiting op een MIDI-synthesizer, maar die uitsluitend de muziekeigenschappen van de computer benutten. Voor de Amiga-gebruiker is er bijvoorbeeld deLuxe Music Construction Set. Dit programma benut de uitstekende stereo kwaliteiten van de Amiga op en top en maakt ook zonder synthesizer de meest fantastische composities.

De toekomst van MIDI

MIDI is op weg naar haar tiende verjaardag en dat zou in de wereld van de snel evoluerende digitale techniek kunnen betekenen dat het al weer bijna verouderd is. We hoeven maar te kijken naar de enorme snelle ontwikkeling van computers die

MIDI nader bekeken

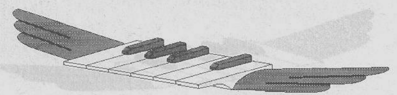
Zonder echt in de diepte te willen gaan en complete te willen zijn is het ongetwijfeld interessant om wat meer te weten te komen van de daadwerkelijke informatie die in de MIDI-boodschap schuilt.
De MIDI-boodschap bevat onder andere informatie over:

- het moment wanneer er een toets wordt aangeslagen;
- de toonhoogte van de toets;
- voor welk kanaal deze toon bedoeld is (n.b. per kanaal kunnen geluiden worden toegekend);
- welk geluid (patch) moet worden aangestuurd;
- of en wanneer dit geluid moet veranderen (program change);
- het volume;
- de snelheid (velocity) waarmee de toets wordt aangeslagen;
- de snelheid waarmee de toets wordt losgelaten;
- wanneer de toest wordt los gelaten;
- de toename of afname van de druk op de toets tijdens het ingedrukt houden ervan (aftertouch);
- de instelling van de effecten (reverb, delay, etc.);
- de waarde van de verschillende controllers.
  Onder controllers worden onder andere verstaan: het modulatiewiel (hiermee kunnen tonen geleidelijk van toonhoogte worden veranderd), de verschillende pedalen (is het pedaal ingedrukt en hoever).

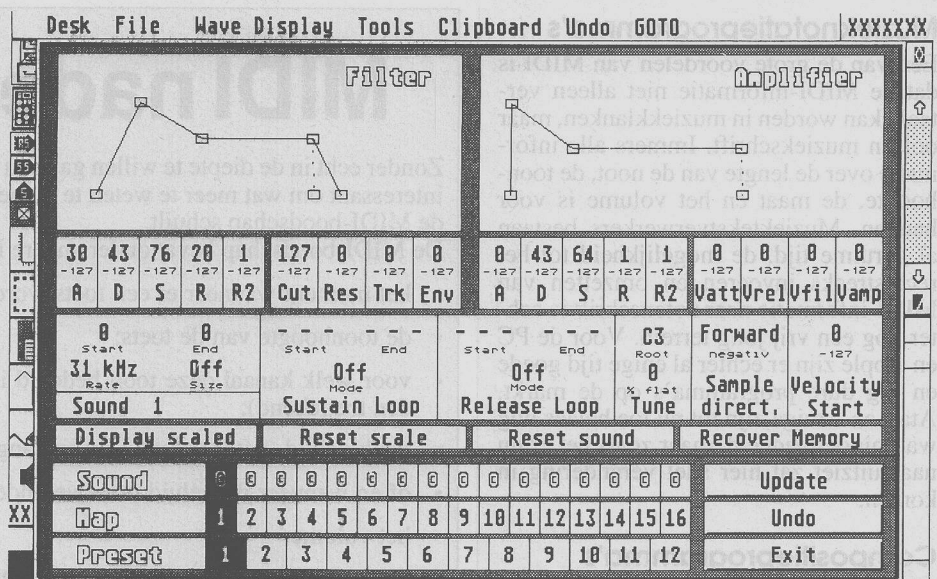
Naast deze informatie, die direct met het bespelen van het instrument samenhangt, wordt er ook nog informatie meegestuurd over een aantal andere zaken die te maken hebben met o.a. timing en stemming.
Een ander belangrijke eigenschap van het MIDI-protocol is dat het net zoals bij een TV werkt met kanalen. In totaal bevat het MIDI-protocol 16 kanalen. Elk kanaal kan volledige MIDI-boodschappen verzenden. Het is aan het ontvangend apparaat om te bepalen welk kanaal of kanalen hij wil ontvangen. Het op elkaar afstemmen van verzenden en ontvangen is dus een uiterst belangrijke zaak, die helaas ook vaak voor veel problemen zorgt.

System Exclusive

In 1983 werden door de verschillende instrumentfabrikanten afspraken gemaakt over de manier waarop deze informatie zou worden gecodeerd. Om de fabrikanten echter toch de nodige vrijheid te geven, heeft het MIDI-protocol veel meer mogelijkheden dan deze overeengekomen standaard. Binnen het MIDI-protocol bestaat er ruimte voor de zogenaamde "system exclusive messages". Dit zijn MIDI-boodschappen die alleen herkend worden door de specifieke instrumenten, sequencers of andere MIDI-randapparatuur van het zelfde merk. Een Yamaha-drumcomputer zal deze system exclusive informatie van een Roland-sequencer niet begrijpen en omgekeerd.



binnen zes jaar tijd van XT, AT, via 386 naar 486-machines ging.
In de tijd dat MIDI onstond, waren er nog niet van die snelle computers met grote geheugens. Het lijkt dus niet ondenkelijk dat MIDI alweer achterhaald zou kunnen zijn. De werkelijkheid is iets genuanceerder. Natuurlijk wordt er momenteel in menig laboratorium hard gewerkt aan de opvolger van MIDI. Hoe de nieuwe MIDI er uit zal komen te zien, is in grote lijnen al bekend.
In de eerste plaats zal de datatransmissie danig versneld worden. Te denken is aan een 10- tot 1000-voud van de huidige snelheid. Deze snelheid is met name nodig om effectief samples (die enorm veel informatie bevatten) te kunnen versturen. (De huidige overdrachtsnelheid bedraagt 31.250 bits per seconde.)
Een andere belangrijke wijziging zal bestaan uit een uitbreiding van het protocol, dat het mogelijk zal maken om ook niet-toets instrumenten makkelijker toegang tot MIDI te verlenen.



Een voorbeeld van een patch-editor programma. Hiermee kunnen bestaande geluiden worden omgevormd tot nieuwe (een patch is een definitie van een synthesizergeluid).

Gelukkig is één van de belangrijkste uitgangspunten bij de nieuw te ontwikkelen MIDI dat het oude protocol ondersteund moeten blijven. Dus u hoeft zich echt geen zorgen te maken of uw programma over twee jaar nog wel zal werken en kan communiceren met de nieuwste klasse synthesizers.

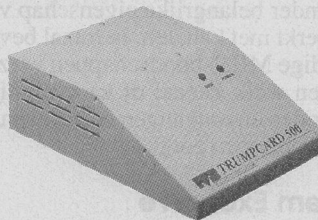
H. Zoete/ U. Schuurmans

Sampling

Sampling is de laatste jaren een begrip geworden in de popmuziek. In de meeste top-40 nummers zijn wel een of meer samples gebruikt. Een sample is niets meer dan een korte digitale opname van een geluid of een muziekinstrument met behulp van en sampler. Dit digitale karakter heeft twee voordelen boven een analoge opname. De kwaliteit van het geluid is aanzienlijk beter en het kan, doordat het in het RAM is opgeslagen heel makkelijk gemanipuleeerd worden. Eén van de meest gebruikte toepassingen van dergelijke samples is de mogelijkheid het opgenomen geluid op verschillende toonhoogtes te kunnen spelen. Men samplet bijvoorbeeld het geluid van een vallend glas en kan vervolgens met dit geluid de melodie van 'Vader Jacob' spelen.
Daarnaast bestaan er nog tal van andere mogelijkheden om het gesampelde geluid te manipuleren. Veel gebruikte opties zijn het achterstevoren afspelen van het geluid en 'versieren' met diverse effecten.
Een groot nadeel van het werken met samples is dat het vastleggen van een geluid ontzettend veel geheugenruimte inneemt. Het probleem is vergelijkbaar met de opslag van gescande kleurenfoto's voor DTP-werk. Iemand die veel met samples werkt zal zich heel wat extra megabytes moeten aanschaffen.

In het volgende nummer:

Harddisk special:



Test van de Trumpcard-harddisks
Deze modellen voor de 500 en 2000 kenmerken zich door de mogelijkheid een aantal partities aan te kunnen brengen, waaronder Mac voor A-Max.

Test van de Vortex-harddisks
Vortex-harddisk worden in Nederland niet geleverd. De distributeur van Vortex bevindt zich in België. Een reden voor Commodore Info om deze drives eens aan een nader onderzoek te onderwerpen.

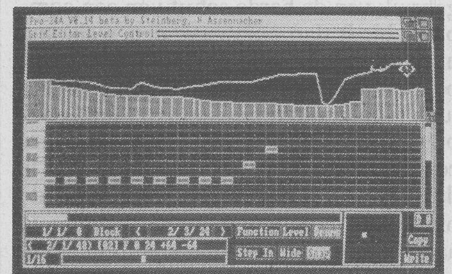
Amiga Action Replay

In nummer 2 van dit jaar testte de redactie 'Nordic Power', een cartridge, die voor alle Amiga-modellen leverbaar is. In nummer 4 een test van Action Replay, dat al eerder voor de C-64 op de markt kwam.

Amiga C development system van Lattice

Programmeren neemt steeds meer mensen in beslag. Lattice is een bekende naam op dit gebied. Uit Amerika ontvingen wij een versie van Lattice C.

Pro Amiga midi-sequencer



Op de CeBIT presenteerde Steinberg de nieuwe sequencer voor de Amiga: Pro 24. Dit programma kwam al eerder uit voor de Atari ST en werd een hit. Wij bekeken deze nieuwe versie voor u.

Vaste rubrieken:

Tips & Trucs 64 met onder andere secret opcodes.
Amiga C over tekst-, teken- en image-functies.